TIME DEPENDENT HOLOGRAPHIC INTERFEROMETRY AND FINITE-ELEMENT ANALYSIS OF HEAT TRANSFER WITHIN A RECTANGULAR ENCLOSURE

Gerald Paul Braun

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIF. 93940

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

TIME DEPENDENT HOLOGRAPHIC INTERFEROMETRY
AND FINITE-ELEMENT ANALYSIS
OF HEAT TRANSFER WITHIN A
RECTANGULAR ENCLOSURE

by

Gerald Paul Braun

September 1976

Thesis Advisor: D. J. Collins

Approved for public release; distribution unlimited.

T174963

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Time Dependent Holographic Interferometry and Finite-Element Analysis of Heat Transfer within a Rectangular Enclosure		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis September 1976
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Gerald Paul Braun		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Air Systems Command Department of the Navy Washington, D. C. 20360		12. REPORT DATE September 1976
		13. NUMBER OF PAGES 195
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Naval Postgraduate School Monterey, California 93940		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Holography
Finite-Element
Convective Heat Transfer

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

In this thesis, the finite-element method was developed to numerically analyze heat transfer by laminar natural convection within a rectangular cavity, a classical fluid flow problem. A second auxiliary case study involving Couette flow was included to test the flexibility of this analysis technique.

Analyzing heat flows experimentally was also explored utilizing holographic interferometry. Specific problems

20. Abstract (cont'd)

encountered during this phase of research are presented with appropriate comments.

DD Form 1473
1 Jan 73
S/N 0102-014-6601

Time Dependent Holographic Interferometry and Finite-Element Analysis of Heat Transfer within a Rectangular Enclosure

by

Gerald Paul Braun
Lieutenant, United States Navy
B.S.E.E., The University of Toledo, 1968

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the
NAVAL POSTGRADUATE SCHOOL
September 1976

Thesis
B80189
c.1

DUDLEY KNOX LIBRARY.
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIF. 93940

ABSTRACT

In this thesis, the finite-element method was developed to numerically analyze heat transfer by laminar natural convection within a rectangular cavity, a classical fluid flow problem. A second auxiliary case study involving Couette flow was included to test the flexibility of this analysis technique.

Analyzing heat flows experimentally was also explored utilizing holographic interferometry. Specific problems encountered during this phase of research are presented with appropriate comments.

TABLE OF CONTENTS

NOMENCLATURE

C_p - specific heat of fluid

D - width of the enclosure

$\triangle$ - area of element triangle

g - gravitational acceleration

Gr_L - Grashof number in L direction = $\dfrac{gBL^3(T_H-T_C)}{\nu^2}$

L - height of the enclosure

L_i - natural coordinates

N_i - interpolation functions

P - pressure (either wall or fluid)

Pr - Prandtl number = $\dfrac{\nu}{\alpha}$

Ra_D - Rayleigh number in D direction = $\dfrac{gBD^3(T_H-T_m)}{\nu}$

Ra_L - Rayleigh number in L direction = $\dfrac{gBL^3(T_H-T_C)}{\nu}$

T - temperature (either wall or fluid)

t - time relative to beginning of solution

T_C - temperature at cold wall

T_H - temperature at hot wall

T_m - mean temperature of fluid = $\dfrac{(T_H+T_C)}{2}$

u - velocity in x-direction

v - velocity in y-direction

x	- independent coordinate in horizontal direction
y	- independent coordinate in vertical direction
α	- thermal diffusivity of fluid = $\frac{\chi}{\rho\,C_p}$
β	- coefficient of thermal expansion of fluid
χ	- thermal conductivity of fluid
μ	- dynamic viscosity of fluid
ν	- kinematic viscosity of fluid = $\frac{\mu}{\rho}$
ρ	- fluid density
$\Omega^{(e)}$	- domain of integration for element (e)

I. INTRODUCTION

A. HOLOGRAPHIC INTERFEROMETRY

The phenomenon of interference has had a considerable influence on the development of physics. Thomas Young's observation and explanation of the interference of the beams through two holes provided the basis for Fresnel's wave theory of light and the same experiment has been used as the foundation of modern coherence theory.

Derived from interference is the technique of interferometry, now one of the important methods of experimental physics. The father of visible-light interferometry was A. A. Michelson, who was awarded in 1907 the Nobel prize in physics for "his optical instruments of precision and the spectroscopic and metrological investigations he has executed with them." Applications to other spectral regions were more recent: the first use of interferometry in radio astronomy was reported in 1947, and infra-red interference spectroscopy was successfully employed some thirteen years later.

Ever since the wave nature of light was generally accepted, interferometry has been the primary method for making measurements with great accuracy. The very small

wavelength of light, on the order of 5×10^{-5}cm, and the fact that interferometric means are available for detecting changes of only a small fraction of this length, indicates the degree of accuracy which can be achieved. The widespread applications of the method attest to its general usefulness. Interferometry is used for testing optical components, optical gauging of machine tools, studying air flow in wind tunnels, and standardizing the fundamental units of length. Therefore it is understandable that any fundamental improvement or innovation in this interferometric technique would find many applications over a wide field.

Holographic interferometry is just such an innovation. Holography may be described as a photographic technique in which the amplitude and phase characteristics emanating from a coherent light source are recorded and later reproduced. This reproduction assumes the form of a three-dimensional image of the original subject. Holography has widened the scope of interferometry to such a degree that holographic interferometry is now considered a standard tool in engineering laboratories all over the world.

Conventional interferometry can be utilized to make measurements on highly polished surfaces of relatively simple shape. Holographic interferometry extends this

range by allowing measurements to be made on three-dimensional surfaces or arbitrary shape and condition. A roughly processed machine part can now be measured to optical tolerance. Furthermore, with the holographic technique a complex object can be examined interferometrically from many different perspectives, because of the three-dimensional nature of the hologram. A single interferometric hologram is equivalent to many observations with a conventional interferometer. This property is especially useful for observations of such things as fluid flow in a wind tunnel. A third departure of holographic interferometry from conventional interferometry is that an object can be interferometrically examined at two different times; one can detect with wavelength accuracy any changes undergone by an object over a period of time. The present object can thus be compared with itself as it was at an earlier time. This is a great advantage in many fields. For example, a large lens can be tested before and after mounting. Similarly, with the aid of pulsed lasers, a machine part can be interferometrically compared with itself statically as well as dynamically.

Methods of holographic interferometry include single- and double-exposure as well as pulsed laser interferometry.

In this thesis, only single-exposure holographic interferometry was considered since it corresponds to real-time interferometry, that is, a method which allows one to observe changes in a subject as they actually occur.

B. CONCEPT AND HISTORY OF THE FINITE-ELEMENT METHOD

One must often resort to numerical procedures in order to obtain quantitative approximate solutions to linear and nonlinear problems in continuum mechanics. However, regardless of the initial assumptions and the methods used to formulate a problem, if numerical methods are employed in evaluating the results, the continuum is, in effect, approximated by a discrete model in the solution process. This observation suggests a logical alternative to the classical approach, namely, represent the continuum by a discrete model at the onset. One such approach, based on the idea of piecewise approximating continuous fields, is referred to as the finite-element method. Its simplicity and generality make it an attractive candidate for applications to a wide range of engineering problems.

Classically, the analysis of continuous systems often began with investigations of the properties of small differential elements of the continuum under investigation. Relationships were established among mean values of various

quantities associated with the infinitesimal elements, and partial differential equations or integral equations governing the behavior of the entire domain were obtained by allowing the dimensions of the elements to approach zero as the number of elements became infinitely large.

In contrast to this classical approach, the finite-element method begins with investigations of the properties of elements of finite dimensions. The equations describing the continuum may be employed in order to arrive at the properties of these elements, but the dimensions of the elements remain finite in the analysis, integrations are replaced by finite summations, and the partial differential equations of the continuous media are replaced, for example, by systems of algebraic or ordinary differential equations. The continuum with infinitely many degrees of freedom is thus represented by a discrete model possessing a finite number of degrees of freedom. Moreover, if certain completeness conditions are satisfied, then, as the number of finite elements is increased and their dimensions are decreased, the behavior of the discrete system converges to that of the continuous system. A significant feature of this procedure is that, in principle, it is applicable to the analysis of finite deformations of materially

nonlinear, nonhomogeneous bodies of any geometrical shape with arbitrary boundary conditions.

The practice of representing a structural system by a collection of discrete elements dates back to the early days of aircraft structural analysis, when wings and fuselages, for example, were treated as assemblages of stringers, skins, and shear panels. By representing a plane elastic solid as a collection of discrete elements composed of bars and beams, Hennikoff [1941] introduced his "framework method," a forerunner to the development of general discrete methods of structural mechanics. Topological properties of certain types of discrete systems were examined by Kron [1939], who developed systematic procedures for analyzing complex electrical networks and structural systems. Courant [1943] presented an approximate solution to the St. Venant torsion problem in which he approximated the warping function linearly in each of an assemblage of triangular elements and proceeded to formulate the problem using the principle of minimum potential energy. Courant's piecewise application of the Ritz method involves all the basic concepts of the procedure now known as the finite-element method. In 1954, Argyris and his collaborators began a series of papers in which they developed certain

generalizations of the linear theory of structures and presented procedures for analyzing complicated discrete structural configurations in forms easily adapted to the digital computer.

The formal presentation of the finite-element method together with the direct stiffness method for assembling elements was attributed to Turner, Clough, Martin, and Topp [1956], who employed the equations of classical elasticity to obtain properties of a triangular element for use in the analysis of plane stress problems. It was Clough [1960], who first used the term "finite elements" in a later paper devoted to plane elasticity problems.

Concepts of the method became more understandable after 1963 when Besseling [1969], Melosh [1970], Fraeys de Veubeke [1971], and Jones [1972] recognized that the finite-element method was a form of the Ritz technique and demonstrated its generality for handling elastic continuum problems. In 1965, the finite-element method received an even broader interpretation when Zienkiewicz and Cheung [1973] reported that it was applicable to all field problems which could be cast into variational form. During the late 1960's and early 1970's, while mathematicians were working on establishing errors, bounds, and convergence

criteria for finite-element approximations, engineers and other appliers of this same method were also studying similar concepts for various problems in the area of solid mechanics.

Although a major portion of the literature written to date on the finite-element method deals with static and dynamic structural analysis, there has been a continuing steady increase in the number of applications in other fields. The goal of this thesis was to develop a computer program, utilizing the finite-element method, which could accurately analyze laminar natural convection within a vertical rectangular enclosure. The program should be able to properly analyze axisymmetric as well as two-dimensional flows.

II. FUNDAMENTAL THEORY OF FINITE-ELEMENT ANALYSIS

In this section the fundamental theory on which the thesis was based is presented. Highlighted topics include the variational principle, some basic concepts of finite-element analysis and the Ritz technique, and finally the method of weighted residuals featuring the Galerkin criterion. The variational principle and the Galerkin method are looked at in detail in regards to the derivation of finite-element equations.

The finite-element method envisions a solution region as built up of many small, interconnected subregions or elements. Such a model of a problem gives a piecewise approximation to the governing equations. The basic premise of the finite-element method is that a solution region can be analytically modeled or approximated by replacing it with an assemblage of discrete elements. These finite-element discretization procedures reduce the problem to one of a finite number of unknowns by dividing the solution region into elements and by expressing the unknown field variable in terms of assumed approximating or interpolation functions within each element. The interpolation functions

are defined in terms of the value of the field variables at specified points called nodes or nodal points. Nodes usually lie on the element boundaries where adjacent elements are considered to be connected. In addition to boundary nodes, an element may also have a few interior nodes (although this was not the case in the choice of linear and quadratic trinagular elements utilized in this thesis). The nodal values of the field variable and the interpolation functions for the elements completely define the behavior of the field variable within the elements. For the finite-element representation of a particular problem, the nodal values of the field variable become the new unknowns. Once these unknowns are found, the chosen interpolation functions define the field variable throughout the assemblage of elements.

Clearly, the nature of the solution and the degree of approximation depend not only on the size and number of the elements used, but also on the interpolation functions selected. As one would expect, functions cannot be arbitrarily chosen since certain compatibility conditions must be satisfied. Often such functions are selected so that the field variable and/or its derivatives are continuous across adjoining element boundaries. Another important

feature of the finite-element method which sets it apart from other approximate numerical methods is its ability to formulate solutions for individual elements before putting them together to represent the entire problem.

The finite-element method has gained much popularity and has been utilized extensively in recent years because it has, in general, several outstanding advantages. These are the following:

1. Non-homogeneous configurations may be treated with relative simplicity.

2. The elements can be graded in size and shape to follow boundaries of arbitrary shape.

3. Once a computer program has been developed, problems of the same variety can be solved simply by supplying the computer with appropriate data.

There are at least three distinct approaches one may employ in order to obtain finite-element equations of a particular system. In order of <u>increasing</u> versatility they are: (1) the direct approach, (2) the variational principle, and (3) the weighted residuals approach.

The direct approach can be used only for relatively simple problems in which discrete elements may be easily identified. Once these elements have been selected, direct

physical reasoning is introduced to establish the element equations in terms of pertinent variables. The final step is then to combine the element equations to form the governing equations of the complete system.

A detailed explanation of the remaining two approaches will be given in Subsections A, B and C to follow.

Whichever one of these three particular approaches is utilized, the finite-element method follows a systematic step-by-step process when applied to continuum problems. They are:

1. Discretize the continuum.

The entire flow region under study is divided into a series of subregions or elements assumed to be interconnected at a finite number of nodal points; thus a program originally exhibiting an infinite number of degrees of freedom is made finite. The elements used can be triangular, rectangular, or almost any shape. Also, information must be fed into a computer giving global coordinates of the nodes and topology of the system.

Finally, selection of which field variables are to be used to satisfactorily describe solution domain must be indicated at this point in the process.

2. Select interpolation functions, $N_i^{(e)}$

From the nodal values one represents the value of the field variable over the element by means of interpolation functions. Often, although not always, polynomials are selected as these functions because they are easy to integrate and differentiate. The number of nodes and the order of the interpolation polynomials are interrelated. The field variable itself may be a scalar, a vector, or a higher-order tensor.

3. Find the element properties.

Essentially, the problem is solved at the element level. The matrix equations expressing the properties of the individual elements are determined. This can be accomplished by any one of the three approaches previously mentioned: the direct method, the variational principle, or the weighted residual method. The approach used depends entirely on the nature of the particular problem.

4. Assemble the element properties to obtain the system equations.

In this step, one combines the matrix equations expressing the behavior of the elements to form the matrix equations expressing the behavior of the entire solution region or system. The matrix equations for the system

exhibit the same form as the equations for an individual element except that they contain many more terms because they include all the nodes. The basis for this assembly procedure stems from the fact that, at a node where elements are interconnected, the value of the field variable is the same for each element sharing that node.

5. Solve the system equations.

From the previous step, a set of simultaneous equations are derived which can now be solved to obtain the unknown nodal values of the field variable. If these equations are linear, a number of standard solution techniques may be employed; if the equations are nonlinear, their solution is more difficult to obtain, but several alternative approaches do exist that lead to satisfactory results.

6. Make additional computations if desired.

It may be desired to use the solution of the system equations to calculate other important parameters, i.e., from the nodal values of the pressure, one might wish to calculate velocity distributions.

It is worth making mention of the fact that several of the steps in the above process are essentially the same regardless of the type of problem (this thesis was devoted

to the fluid mechanics problem). Thus, only steps three (3) and six (6) might differ for any given situation, in that the equations describing the elements could vary. The other steps would be the same. This generality of the finite-element method is, without doubt, one of its greatest strengths.

A. VARIATIONAL PRINCIPLE

Often, continuum problems have different but yet equivalent formulations, such as a differential formulation and a variational formulation. In the differential case, the problem is to integrate a differential equation or a system of differential equations subject to given boundary conditions. In the classical variational formulation, the problem is to find the unknown function or functions which extremize (maximize, minimize) or make stationary a functional or system of functionals subject to the same specified boundary conditions. The two problem formulations are equivalent because the functions that satisfy the differential equations and their boundary conditions also extremize or make stationary the functionals. This equivalence is apparent from the calculus of variations, which shows that the functionals are extremized or made stationary only when

one or more Euler equations and their boundary conditions are satisfied. Consequently, these equations are precisely the governing differential equations of the problem. To illustrate this duality concept, Appendix B provides a brief review and introduction to some basic ideas of the calculus of variations.

B. FINITE-ELEMENT METHOD AND THE RITZ TECHNIQUE

The Ritz technique is basically a procedure for transforming a continuous medium into an approximated lumped parameter system. A more qualitative definition would be that the Ritz method consists of <u>assuming</u> the form of the unknown adjustable parameters. From this family of trial or coordinate functions, that particular function which renders the functional stationary is then selected. The procedure is to substitute the trial functions into the functional and thereby express the functional in terms of the adjustable parameters. This functional is then differentiated with respect to each parameter, and the resulting equation is set equal to zero. If there are n unknown parameters, there will be n simultaneous equations to be solved for these parameters. By this means, the approximate solution is chosen from the family of assumed solutions.

This procedure does nothing more than give one the "best" solution from the family of assumed solutions. Clearly, then, the accuracy of the approximate solution depends on the choice of trial functions. These trial functions are required to be defined over the whole solution domain and must satisfy at least some and usually all of the boundary conditions. Sometimes, if the general nature of the desired solution is known, the approximation can be improved by choosing the trial functions to reflect this nature. If, by chance, the exact solution is contained in the family of trial solutions, then the Ritz technique gives the exact solution as expected. Generally, the approximation improves as the size of the family of trial functions and the number of adjustable parameters increase. If the trial functions are part of an infinite set of functions that are capable of representing the unknown function to any degree of accuracy, the process of including more and more trial functions leads to a series of approximate solutions which converge to the true solution. Often a family of trial functions is constructed from polynomials of successively increasing degree.

To illustrate the Ritz technique, consider the following simple example. Suppose it is desired to find the general

function $\phi(x)$ satisfying

$$\frac{d^2\phi}{dx^2} = -f(x)$$

with boundary conditions of $\phi(a)=A$ and $\phi(b)=B$ specified. It is assumed that $f(x)$ is a continuous function in the closed interval $[a,b]$. This problem is equivalent to finding the function $\phi(x)$ that minimizes the functional

$$I(\phi) = \int_b^a \left[\tfrac{1}{2}\left(\frac{d\phi}{dx}\right)^2 - f(x)\phi(x) \right] dx$$

which is of the form $I(\phi) = \int_{x_1}^{x_2} F(x,\phi,\ \phi_x,\ f(x))dx$

Ignoring the fact that this problem possesses an exact solution, we will attempt to find an approximate solution. According to the Ritz method, the desired solution can be assumed to be approximately represented in $[a,b]$ by a combination of selected trial functions of the form

$$\phi(x) = C_1\,\psi_1(x)+C_2\,\psi_2(x)+\ldots+C_n\,\psi_n(x)\ , \quad a \leq x \leq b$$

where the n constants C_i are the adjustable parameters to be determined. The trial functions should be selected so that the expression for $\phi(x)$ satisfies the boundary conditions regardless of the choice of the constants C_i. Using

polynomials is a simple and convenient way of constructing the trial functions. Therefore

$$\phi(x) \approx (x-a)(x-b)(C_1+C_2x+C_3x^2+\dots+C_nx^{n-1})$$

is a possible series of trial functions. When this approximate expression for $\phi(x)$ is substituted into the functional to be minimized, and after the integration has been carried out, the functional is of the form

$$I = I(C_1,\ C_2,\ ----,\ C_n).$$

Since the C_i are required to be chosen such that they minimize I, employing differential calculus, the following partial differential equations are formulated

$$\frac{\partial I}{\partial C_1} = 0,\ \frac{\partial I}{\partial C_2} = 0,\ \dots,\ \frac{\partial I}{\partial C_n} = 0$$

These n equations are then solved for the n parameters C_i, and the accuracy of the approximate solution depends on the number of C's used in the trial function. Generally, as n increases the accuracy improves. To assess the improvement in accuracy as more C's are utilized, the problem is solved repeatedly by taking successively more terms in the approximation, that is

$$\phi_1(x) \approx (x-a)(x-b)C_1$$

$$\phi_2(x) \approx (x-a)(x-b)(C_1+C_2x)$$

$$\phi_3(x) \approx (x-a)(x-b)(C_1+C_2x+C_3x^2)$$

and so on. By comparing the results at the end of each calculation, the effect on accuracy of adding more terms can be estimated.

The finite-element method and the Ritz technique are essentially equivalent. Each method uses a set of trial functions as the starting point for obtaining an approximate solution; both methods take linear combinations of these trial functions; and both methods seek the combination of trial functions that renders a given functional stationary. The major difference between these approximating methods stems from the fact that the assumed trial functions in the finite-element method are not defined over the entire solution domain, and they must satisfy not just any boundary conditions, but only certain continuity conditions and then only sometimes. Since the Ritz technique uses functions construed over the whole domain, it can be employed only for domains of relatively simple geometric shape. Also, these trial functions associated with the Ritz method are required to satisfy at least some and usually all of the

boundary conditions. In the finite-element method the same geometric limitations exist, but only for the elements. Due to the fact that elements with simple shapes can be assembled to represent exceedingly complex geometries, the finite-element method is a far more versatile tool than the Ritz technique. From a strict mathematical standpoint, the finite-element method is a special case of the Ritz technique only when the piecewise trial functions obey certain continuity and completeness conditions that are stipulated over just the element alone.

C. METHOD OF WEIGHTED RESIDUALS (GALERKIN'S METHOD)

The third and final approach to the finite-element method involves a procedure that is more generalized and straightforward than either of its two predecessors.

The relationship between the well-known Ritz technique and the finite-element method enables one to view the finite-element discretization procedure as simply another means for finding approximate solutions to variational problems. In fact, these finite-element equations were shown to be derived by requiring that a given functional be stationary. This broad variational interpretation is the one most widely used to derive element equations, and it is the

most convenient approach whenever a classical variational statement exists for a given problem.

However, applied scientists and engineers encounter practical problems for which classical variational principles are unknown. In these cases finite-element techniques are still applicable, but more generalized procedures characteristic of the method of weighted residuals must be employed to derive the element equations. Through certain generalizations, finite-element equations may be derived directly from the governing differential equations of the problem without reliance on any classical, quasi-variational, or restricted variational "principles." This procedure allows one to apply the finite-element method to almost all practical problems of mathematical physics.

The method of weighted residuals is a technique for obtaining approximate solutions to linear and nonlinear partial differential equations. It offers still another means with which to formulate the finite-element equations. Applying the method of weighted residuals involves basically two steps. The first step is to assume the general functional behavior of the dependent field variable in some way so as to approximately satisfy the given differential equation along with its associated boundary conditions.

Substitution of this approximation into the original differential equation and boundary conditions then results in some error called a residual. This residual is required to vanish in some average sense over the entire solution domain. The second step entails solving the equation(s) resulting from step one and thereby specializing the general functional form to a particular function, which in turn becomes the approximate solution sought.

To be more specific, the following typical problem is offered. Suppose it is desired to find an approximate functional representation for a general field variable ϕ governed by the differential equation

$$\mathcal{L}(\phi) - f = 0 \tag{2.1}$$

in the domain D bounded by the surface Σ. $\mathcal{L}$ is a linear or nonlinear differential operator and the function f is a known function of the independent variables. Also, proper boundary conditions are assumed to be prescribed on Σ. The method of weighted residuals is now applied in two steps. First, the unknown exact solution ϕ is approximated by $\hat{\phi}$, where either the functional behavior of $\hat{\phi}$ is completely specified in terms of unknown parameters, or the functional dependence on all but one of the independent variables is given while the functional dependence on the

remaining independent variable is left unspecified. Thus the dependent variable is approximated by

$$\phi \approx \hat{\phi} = \sum_{i=1}^{m} N_i C_i \tag{2.2}$$

where the N_i are the assumed functions and the C_i are either the unknown parameters or unknown functions of one of the independent variables. The m functions N_i are usually chosen to satisfy the global boundary conditions of the system in question. When $\hat{\phi}$ is substituted into equation 2.1, it is unlikely that this equation will not be satisfied, that is,

$$\mathcal{L}(\hat{\phi}) - f \neq 0$$

but in fact,

$$\mathcal{L}(\hat{\phi}) - f = e$$

where e is the residual or error that results from approximating ϕ by $\hat{\phi}$. The method of weighted residuals seeks to determine the m unknowns C_i in such a way that the error e over the entire solution domain is small. This is accomplished by forming a weighted average of the error and specifying that this weighted average vanish over the solution domain. In other words, m linearly independent weighting functions, W_i, are chosen such that

$$\int_D \left[\mathcal{L}(\hat{\phi}) - f\right] W_i \, dD = \int_D e W_i \, dD = 0, \quad i=1,2,\text{---},m \tag{2.3}$$

The form of the error distribution principle expressed in equation 2.3 depends on the choice of weighting functions. Once these are specified, equation 2.3 represents of a set of m equations, which may be either algebraic or ordinary differential. The second step is to solve for the C_i's and hence obtain an approximate representation of the unknown general field variable $\emptyset$ via equation 2.2. There are many linear problems and even some nonlinear problems for which it can be shown that, as $m \rightarrow \infty$, $\hat{\emptyset} \rightarrow \emptyset$, but, in general, studies of convergence and error bounds are scarce.

Due to the broad choice of weighting functions or error distribution principles than can be used, a variety of weighted residual techniques are likewise available. The error distribution principle most often utilized to derive finite-element equations in the field of aeronautics is known as the Galerkin criterion, or Galerkin's method. Here, the weighting functions are chosen to be the same as the approximating functions employed to represent $\emptyset$, that is, $W_i=N_i$ for $i=1,2,\ldots.,m$. Therefore Galerkin's method requires that

$$\int_D \left[\mathcal{L}(\hat{\emptyset})-f\right] N_i dD = 0 \qquad (2.4)$$

In the preceding section pertaining to the Ritz technique,

it was assumed that the entire solution domain was being dealt with. However, because equation 2.1 holds for any point in this region, it also holds for any collection of points defining an arbitrary subdomain or element of the whole domain. Consequently, attention may be focused directly on an individual element by means of a local approximation analogous to equation 2.2, but being defined as valid for only one element at a time. Now the finite-element representations of a general field variable become available. The functions N_i become what are known as the interpolation functions $N_i^{(e)}$ defined over the element, and the C_i are the undetermined parameters, which may be the nodal values of the field variable or its derivatives. Then, from Galerkin's method, the equations governing the behavior of an element of the solution domain may be written as

$$\int_D \left[\mathcal{L}(\phi^{(e)})-f^{(e)}\right] N_i^{(e)} dD^{(e)} = 0, \quad i=1,2,\ldots,r \qquad (2.5)$$

where, as before, the superscript (e) restricts the range to one element, and

$\phi^{(e)} = \lfloor N^{(e)} \rfloor \{\phi\}^{(e)}$

$f^{(e)}$ = forcing function defined over element (e)

r = number of unknown parameters assigned to the element.

There exists a set of equations similar to equation 2.5 for each element of the whole assemblage. Prior to assembling the system equations from the individual element equations, it is required that the choice of approximating functions N_i guarantee the interelement continuity along the boundary necessary for the assembly process. If the field variable is continuous at element interfaces, then C^o continuity exists; if, in addition, first derivatives of the variable are continuous, C^1 continuity is said to occur; if second derivatives are also continuous, a region of C^2 continuity exists; and so on. This is the standard definition and notation utilized for expressing the degree of continuity of a field variable at element junctions. The higher the order of continuity required in the solution, the narrower one's choice of interpolation functions becomes.

With the above definition of continuity in mind, the compatibility and completeness requirements for such interpolation functions may be stated. If the functions appearing under the integrals in the element equations contain derivatives up to the (n+1)th order, then the following stipulations must be satisfied for assurance of convergence as the element size decreases.

<u>Compatibility</u> requirement: At element interfaces, C^n continuity must exist.

<u>Completeness</u> requirement: Within an element, C^{n+1} continuity must exist. These requirements hold regardless of whether the element equations (integral expressions) were derived using the variational technique or the Galerkin methed. For this thesis, n was taken to have a value of zero.

Integration by parts is a convenient way to introduce the natural boundary conditions that must be satisfied on some portion of the system exterior or boundary. Although the boundary terms containing these imposed conditions appear in the equations for each element, during the assembly of the element equations only the boundary elements give nonvanishing contributions. After the assembly process has been completed, the fixed boundary conditions (i.e., specified velocity, pressure or temperature) are conveniently introduced to help simplify the final matrix form of the finite element equation.

III. ANALYSIS OF CONVECTIVE HEAT TRANSFER BETWEEN PARALLEL PLATES

The transfer of heat energy across a fluid layer is accomplished, in general, through the mechanisms of conduction, convection and radiation. This last phenomenon is usually a function of the fluid enclosed between the surfaces and the nature, temperature and configuration of the enclosing boundaries. Radiation takes place independently of the conduction and convection as long as there is no absorption by the fluid, and therefore under these conditions it can be considered separately. The phenomena of conduction and convection are closely interdependent and are usually analyzed together. Buoyancy forces result from differences in density within the fluid and are caused by heat transfer to or from this fluid. Natural convection may then be thought of as fluid motion of the system due to the activation of these buoyance forces. In a two-dimensional plane, such heat transfer across a vertical, enclosed fluid layer is a function of the Grashof number, the Prandtl number and the fluid layer height-to-width ratio (L/D).

Natural convection plays a very important role in materials processing at high temperatures where agitation by other means is impracticable, or where the existence of temperature gradients is an inherent characteristic of the system.

The steady convective motion of a lubricating fluid contained within a long, rectangular enclosure was investigated. Holographic interferometry and numerical approximation were the experimental and theoretical analysis tools, respectively.

The two vertical walls of the enclosure were held at different temperatures, and the top and bottom were deemed perfect insulators (Figure 1). It was considered that the length of the enclosure (7 inches) was sufficiently long in the direction normal to the plane of Figure 1 for the motion to be assumed two-dimensional. Another assumption made was that the fluid motion was laminar. Experimental evidence indicates that such an assumption is valid provided the Rayleigh number based on cavity height is less than about 10^8 (Ra_L in this study was calculated to be 1.018×10^7). Using this value and a value of the Prandtl number of 1.0755×10^4, determined from the ratio of kinematic viscosity to thermal diffusivity of the fluid, a system

Grashof number of 946.4 was calculated. The temperatures of the vertical walls x=0 and x=D were defined to be T_H and T_C respectively. If (T_H-T_C) in degrees Fahrenheit is sufficiently small with respect to T_C, the Boussinesq approximation may be introduced which neglects density variations in inertia terms of the equations of motion, but retains it in the buoyancy term. One final assumption was made that all other relevant thermodynamic and transport properties were independent of temperature and that compressibility and viscous dissipation effects were negligible.

The problem now was to find the time and spatial dependence of the velocities and the temperatures within the system.

The governing differential equations expressing conservation of mass, momentum (both in x- and y-directions) and energy were

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{3.1}$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = \frac{\mu}{\rho}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) - \frac{1}{\rho}\frac{\partial P}{\partial x} + gB(T-T_m) \tag{3.2}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = \frac{\mu}{\rho}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) - \frac{1}{\rho}\frac{\partial P}{\partial y} \tag{3.3}$$

$$\frac{\partial T}{\partial t} + u\frac{\partial T}{\partial x} + v\frac{\partial T}{\partial y} = \alpha \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right) \qquad (3.4)$$

The solution to the foregoing set of dynamic equations must satisfy the following boundary conditions on the walls,

$u_o = v_o = 0$,	no velocity on any of the four walls
$T = T_H$ or T_C	given on the two vertical walls
$P = P_{ATMOS}$,	also given on the two vertical walls.

IV. THEORETICAL RESULTS

A. FINITE-ELEMENT ANALYSIS OF THE CONTINUUM

1. Discretization of the Continuum

Since the fundamental premise of the finite-element method is that a continuum or solution domain of arbitrary shape can be accurately modeled by an assemblage of simple shapes, most finite elements are geometrically simple also. This statement especially pertains to the choice of the triangular-shaped element which would represent the unknown system parameters in this study, that is, the velocity, temperature and pressure. The main reason behind this choice was the fact that the three-node flat triangular element is the simplest two-dimensional element available, and hence an assemblage of triangles could always depict a two-dimensional domain with any number of straight sides. The solution domain in this problem was the vertical rectangular enclosure, a relatively simple-shaped continuum which posed no problem for the triangular elements. Twelve (12) elements were utilized to represent the 8.5 inch by 1.875 inch area. They were interconnected to each other and the boundary at a total of thirty-five (35) nodal points, of which twelve (12) were corner nodes (Figure 2).

Arriving at this figure of thirty-five nodes was not an arbitrary process. Once pressure was chosen to be linearly approximated, system velocities and temperature were required to assume polynomial approximation of one degree higher, or quadratic, if the highest solution accuracy was to be achieved. For triangular elements, a complete n th-order polynomial requires $\frac{1}{2}(n+1)(n+2)$ nodes for its specification. Therefore, a 1st order, or linearly approximated, polynomial is associated with a three-node triangle; and a quadratic polynomial relates to a six-node triangle.

The three-node elements, with their nodes on the corners, may be thought of as being superimposed onto the six-node elements. Such elements contain, in addition to the corner nodes, nodes located at the midpoint of each side of the triangle. Twelve triangular elements of the six-node variety may be interconnected to form the solution domain shown in Figure 2; this domain possessing exactly thirty-five nodes.

Each element (6-node and 3-node) specifies uniquely a complete polynomial of the order necessary to give C^0 continuity, and hence satisfy the completeness and compatibility requirements for elemental assemblage.

Next, the distinction between local and global node-numbering had to be made. Since each element in the triangular mesh had six nodes, the local nodes were identified as such by starting in the upper left hand corner of each element and numbering counterclockwise around the element. The global node system is a method for uniting these independent elements along with their nodes into one distinct entity. Figure 3 summarizes the relation between local and global numbering for four (4) such elements. This figure defines the system topology or the connectivity of the system.

2. Selection of the Interpolation Functions

In the preceding subsection it was mentioned that linear approximation was used for values representing nodal pressures, while both velocity and temperature varied in a quadratic fashion within the elements. Such a relationship was based on the governing equations of the system, in which the highest order of partial differential equations involving pressure was one, while partial derivatives of u, v and T existed up to second order. Therefore, choosing linear pressures required the remaining three nodal parameters or field variables to take on quadratic approximation.

The functions employed to represent the behavior of these field variables within an element are known as interpolation or approximating functions. Their order within an element depends on the number of degrees of freedom assigned to that element. In this study, two different polynomial series were selected as the first and second order interpolation functions. Associated with these series were coefficients made up of generalized coordinates, that is, independent parameters which specified the magnitude of the prescribed distribution for each field variable (u, v, P, T). These polynomials were represented as follows

$$P(x,y)^{(e)}=C_1^{(e)}+C_2^{(e)}x+C_3^{(e)}y \tag{4.1}$$

for the linear pressure terms, and

$$\begin{aligned}\phi(x,y)^{(e)}=&C_1^{(e)}+C_2^{(e)}x+C_3^{(e)}y+C_4^{(e)}x^2+\\ &C_5^{(e)}xy+C_6^{(e)}y^2\end{aligned} \tag{4.2}$$

with ϕ being a generalized quadratic field variable (either u, v, or T in this case, and the superscript 'e' standing for element.

The next step in the process was to solve for the generalized coordinate $C_i^{(e)}$ in terms of the as yet unknown field variables. This gave the desired interpolation, but the form of the resulting equations was not convenient. As a final step then, the equations were rearranged until they appeared as

$$P(x,y)^{(e)}=N_1^P(x,y)P_1+N_2^P(x,y)P_2+ N_3^P(x,y)P_3=\lfloor N^P \rfloor \{P\} \tag{4.3}$$

and

$$\phi(x,y)^{(e)}=N_1^\phi(x,y)\phi_1+N_2^\phi(x,y)\phi_2+N_3^\phi(x,y)\phi_3+ N_4^\phi(x,y)\phi_4+N_5^\phi(x,y)\phi_5+N_6^\phi(x,y)\phi_6=\lfloor N^\phi \rfloor\{\phi\} \tag{4.4}$$

where N_i^u, N_i^v, and N_i^T were the specific interpolation functions in equation (4.4) for this study and were all equal in form, i.e., $N_i^u=N_i^v=N_i^T=N_i$.

3. Determination of the Elemental Properties

In this thesis, the Galerkin method was utilized to determine the element properties. This procedure applied at a general node i of an isolated element becomes, in view of equations 3.1-3.4,

$$\int_{\Omega^{(e)}} H_i\left(\frac{\partial u^{(e)}}{\partial x} + \frac{\partial v^{(e)}}{\partial y}\right) dxdy=0 \qquad (4.5)$$

$$\int_{\Omega^{(e)}} W_i\left[\frac{\mu}{\rho}\left(\frac{\partial^2 u^{(e)}}{\partial x^2} + \frac{\partial^2 u^{(e)}}{\partial y^2}\right) - \frac{1}{\rho}\frac{\partial P^{(e)}}{\partial x} + gB(T^{(e)}-T_m) - u^{(e)}\frac{\partial u^{(e)}}{\partial x} - v^{(e)}\frac{\partial u^{(e)}}{\partial y} - \frac{\partial u^{(e)}}{\partial t}\right] dxdy=0 \qquad (4.6)$$

$$\int_{\Omega^{(e)}} W_i\left[\frac{\mu}{\rho}\left(\frac{\partial^2 v^{(e)}}{\partial x^2} + \frac{\partial^2 v^{(e)}}{\partial y^2}\right) - \frac{1}{\rho}\frac{\partial P^{(e)}}{\partial y} - u^{(e)}\frac{\partial v^{(e)}}{\partial x} - v^{(e)}\frac{\partial v^{(e)}}{\partial y} - \frac{\partial v^{(e)}}{\partial t}\right] dxdy=0 \qquad (4.7)$$

$$\int_{\Omega^{(e)}} W_i\left[\alpha\left(\frac{\partial^2 T^{(e)}}{\partial x^2} + \frac{\partial^2 T^{(e)}}{\partial y^2}\right) - u^{(e)}\frac{\partial T^{(e)}}{\partial x} - v^{(e)}\frac{\partial T^{(e)}}{\partial y} - \frac{\partial T^{(e)}}{\partial t}\right] dxdy=0 \qquad (4.8)$$

where $W_i(x,y)$ and $H_i(x,y)$ are the weighting or interpolation functions, which were taken as

$$W_i=N_i \text{ and } H_i=N_i^P=L_i \text{ (natural coordinates).}$$

The inertia terms in equations 4.6, 4.7 and 4.8 considerably increased the degree of difficulty of this fluid flow problem when compared to an incompressible viscous flow without inertia. This is because the above mentioned equations are nonlinear, thereby forcing an iterate procedure to be introduced and repeated until the $u_{n+1}^{(e)}$, $v_{n+1}^{(e)}$, and $T_{n+1}^{(e)}$

values converged to the previous $u_n^{(e)}$, $v_n^{(e)}$, and $T_n^{(e)}$ solutions. The subscript n runs from zero to some positive number at which the field variable passes a convergence test.

Integrating each term of equations 4.5-4.8 by parts, and making use of the approximations of equations 4.3 and 4.4, the following results on an elemental level were obtained

$$\int_{\Omega^{(e)}} (N_i^P \frac{\partial \lfloor N \rfloor}{\partial x} \, dxdy) \left\{ u \right\} + \int_{\Omega^{(e)}} (N_i^P \frac{\partial \lfloor N \rfloor}{\partial y} \, dxdy) \left\{ v \right\} = 0 \qquad (4.9)$$

$$\int_{\Omega^{(e)}} \frac{\mu}{\rho} (\frac{\partial N_i}{\partial x} \frac{\partial \lfloor N \rfloor}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial \lfloor N \rfloor}{\partial y}) dxdy \left\{ u \right\} - \frac{1}{\rho} \int_{\Omega^{(e)}} \frac{\partial N_i}{\partial x} \lfloor N^P \rfloor \, dxdy \left\{ P \right\} - gB \lfloor N_i \rfloor \left\{ T \right\} - \int_{\Omega^{(e)}} (N_i \lfloor N \rfloor \, dxdy) \left\{ \frac{\partial u}{\partial t} \right\} = \left\{ gBT_m \right\} + \int_C N_i X^* ds \qquad (4.10)$$

$$\int_{\Omega^{(e)}} \frac{\mu}{\rho} (\frac{\partial N_i}{\partial x} \frac{\partial \lfloor N \rfloor}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial \lfloor N \rfloor}{\partial y}) dxdy \left\{ v \right\} - \frac{1}{\rho} \int_{\Omega^{(e)}} \frac{\partial N_i}{\partial y} \lfloor N^P \rfloor \, dxdy \left\{ P \right\} - \int_{\Omega^{(e)}} (N_i \lfloor N \rfloor \, dxdy) \left\{ \frac{\partial v}{\partial t} \right\} = \int_C N_i Y^* ds \qquad (4.11)$$

$$\int_{\Omega^{(e)}} \alpha \left(\frac{\partial N_i}{\partial x} \frac{\partial \lfloor N \rfloor}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial \lfloor N \rfloor}{\partial y}\right) dxdy \left\{ T \right\}$$

$$- \int_{\Omega^{(e)}} (N_i \lfloor N \rfloor \, dxdy) \left\{ \frac{\partial T}{\partial t} \right\} = \int_C N_i Z^* ds \tag{4.12}$$

where $N_i X^*ds$, $N_i Y^*ds$ and $N_i Z^*ds$ are simply lumped-sum contour integrals that introduce the natural boundary conditions for u, v and T respectively. These integral values were labeled QX, QY, and QZ in the computer program. The last term on the left hand side of equations 4.10-4.12 represents the transient nature of the system.

Finally, the element matrix equations were written by inspection from equations 4.9-4.12 and were of the general form

$$[K]^{(e)} \left\{ \emptyset \right\}^{(e)} - [K_t]^{(e)} \left\{ \dot{\emptyset} \right\}^{(e)} = \left\{ R \right\}^{(e)} \tag{4.13}$$

where the square matrices $[K]^{(e)}$ AND $[K_t]^{(e)}$ are known as stiffness matrices, the column vectors $\left\{ \emptyset \right\}^{(e)}$ and $\left\{ \dot{\emptyset} \right\}^{(e)}$ are the nodal field variable and time derivative vectors, respectively. The column vector $\left\{ R \right\}^{(e)}$ signifies the resultant nodal force vector for the element. In the actual computer program, the following identities were used

$$[K] = [TM], \; [K_t] = [CD], \; \left\{ \emptyset \right\} = \left\{ X \right\}, \; \left\{ \dot{\emptyset} \right\} = \left\{ \dot{X} \right\},$$

$$\text{and } \left\{ R \right\} = \left\{ RHS \right\}$$

and the element matrix equations were

$$
\begin{array}{cc}
(3r+s)\times(3r+s) & (3r+s)\times 1 \\
\left[\begin{array}{c:c:c:c}
\nu[K_1] & [0] & -\frac{1}{\rho}[K_2]^T & g\beta[I] \\ \hdashline
[0] & \nu[K_1] & -\frac{1}{\rho}[K_3]^T & [0] \\ \hdashline
-[K_2] & -[K_3] & [0] & [0] \\ \hdashline
[0] & [0] & [0] & \alpha[K_1]
\end{array}\right] &
\left[\begin{array}{c}
\{u\}^e \\ \hdashline \{v\}^e \\ \hdashline \{P\}^e \\ \hdashline \{T\}^e
\end{array}\right]
\end{array}
\;-
$$

$$
\begin{array}{ccccc}
\left[\begin{array}{c:c:c:c}
[CD] & [0] & [0] & [0] \\ \hdashline
[0] & [CD] & [0] & [0] \\ \hdashline
[0] & [0] & [0] & [0] \\ \hdashline
[0] & [0] & [0] & [CD]
\end{array}\right] &
\left[\begin{array}{c}
\{\dot{u}\}^e \\ \hdashline \{\dot{v}\}^e \\ \hdashline \{\dot{P}\}^e \\ \hdashline \{\dot{T}\}^e
\end{array}\right] & = &
\left[\begin{array}{c}
\{AX\}^e \\ \hdashline \{QY\}^e \\ \hdashline \{QZC\}^e \\ \hdashline \{QZ\}^e
\end{array}\right] \\
(3r+s)\times(3r+s) & (3r+s)\times 1 & & (3r+s)\times 1
\end{array}
$$

where, $\{AX\}^e = \{g\beta T_M + QX\}$

In the above assemblage, the individual matrix notation utilized was

$$\left[K_1\right] = K_1(i,j) = \int_{\Omega^e} \left(\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y}\right) dxdy$$

$$\left[K_2\right]^T = K_2^T(i,j) = \int_{\Omega^e} \left(\frac{\partial N_i}{\partial x} Nj^P\right) dxdy$$

$$\left[K_3\right]^T = K_3^T(i,j) = \int_{\Omega^e} \left(\frac{\partial N_i}{\partial y} N_j^P\right) dxdy$$

$$\left[K_2\right] = K_2(i,j) = \int_{\Omega^e} \left(\frac{\partial N_j}{\partial x} N_i^P\right) dxdy$$

$$\left[K_3\right] = K_3(i,j) = \int_{\Omega^e} \left(\frac{\partial N_j}{\partial y} N_i^P\right) dxdy$$

$$\left[CD\right] = CD(i,j) = \int_{\Omega^e} N_i N_j dxdy .$$

Also, gBT_m is the u velocity forcing function, and r and s are the number of nodes where velocity (or temperature) and pressure are interpolated at, respectively. In this study, r=6 and s=3, therefore the element matrices were 21x21 and the element column vectors 21x1.

Once the matrix equations were compiled or assembled on the element level, assembling these properties to obtain the system equations in matrix form also was a relatively simple operation for the digital computer. In essence, the large square matrices were derived by systematically adding

together the contributions of each individual element matrix, and inserting prescribed nodal variables or boundary conditions where applicable. As was brought out in the theoretical section of this thesis, the final assembly now became a system of ordinary differential equations resembling the same format as equation 4.13, i.e.

$$\left[K \right] \left\{ \phi \right\} - \left[K_t \right] \left\{ \dot{\phi} \right\} = \left\{ R \right\} \tag{4.14}$$

The problem solution was completed when these equations were solved for the nodal parameters $\left\{ \phi \right\}$, subject to the discretized initial conditions.

B. DERIVATION OF ELEMENT MATRICES

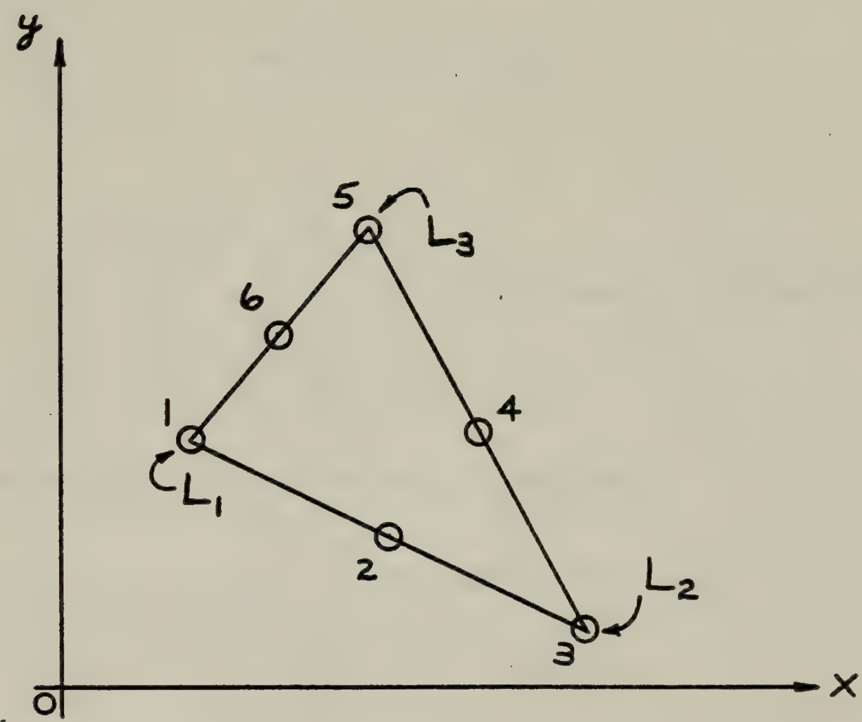
Derivation of various element matrices, referred to previously as simply area integrals over the solution domain, will be discussed in this section. The evaluation of each matrix will be in terms of natural coordinates, that is, weighting functions relating the coordinates of the end nodes to the coordinate of any interior point belonging to the element. The weighting functions are not independent of one another, since their sum must equal unity, i.e.

$$\sum_{i=1}^{n} L_i = 1 \tag{4.15}$$

where n is the number of external nodes of the element. This expression can be interpreted to mean that one and only one coordinate is associated with node i, having a unit value there and a zero value at every other node. As was previously mentioned in other sections, a general triangular shaped element, such as sketched below, was employed. Then by equation 4.15

$$L_1+L_2+L_3 = 1$$

A cartesian coordinate system is used since the fluid flow is assumed to be two-dimensional. Similar results could be derived using cylindrical coordinates for an axisymmetrically shaped element. The original Cartesian coordinates of a



point in the element can now be linearly related to the new natural coordinates by the equations

$$x = L_1x_1+L_2x_2+L_3x_3 \tag{4.16}$$

and

$$y = L_1y_1+L_2y_2+L_3y_3 \tag{4.17}$$

Solving for the natural coordinates in terms of the Cartesian coordinates gives

$$L_1(x,y) = \frac{1}{2\Delta}(a_1+b_1x+c_1y) \tag{4.18a}$$

$$L_2(x,y) = \frac{1}{2\Delta}(a_2+b_2x+c_2y) \tag{4.18b}$$

and finally

$$L_3(x,y) = \frac{1}{2\Delta}(a_3+b_3x+c_3y) \tag{4.18c}$$

where

$$2\Delta = \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} = 2\ (\text{area of triangle 1-2-3})$$

$$a_1=x_2y_3-x_3y_2,\ b_1=y_2-y_3,\ c_1=x_3-x_2$$

$$a_2=x_3y_1-x_1y_3,\ b_2=y_3-y_1,\ c_2=x_1-x_3$$

$$a_3=x_1y_2-x_2y_1,\ b_3=y_1-y_2,\ c_3=x_2-x_1$$

The interpolation functions N_i for the linear pressures in terms of natural coordinates are merely

$$N_1^P = L_1, \; N_2^P = L_2, \; N_3^P = L_3$$

but those interpolation functions that relate to the velocities and temperatures stemming from quadratic approximation possess the form

$$\begin{aligned}
N_1 &= 2L_1^2 - L_1 \\
N_2 &= 4L_1L_2 \\
N_3 &= 2L_2^2 - L_2 \\
N_4 &= 4L_2L_3 \\
N_5 &= 2L_3^2 - L_3 \\
N_6 &= 4L_1L_3
\end{aligned} \tag{4.19}$$

Another way of envisioning $L_i(x,y)$ for the triangular element is to consider it a ratio of areas. Figure 4 shows how the natural coordinates, often called area coordinates, are related to areas. In this figure, when the point (x_p,y_p) is located on the boundary of the element, one of the area segments vanishes and hence the appropriate area coordinate along that particular boundary is identically zero. For example, if (x_p,y_p) is on line 1-2, then

$$L_3 = \frac{A_3}{\Delta} = 0 \quad \text{since } A_3 = 0$$

There is also a convenient analytical method for integrating area coordinates over the area of a triangular element and involves the formula

$$\int_{A^{(e)}} L_1^{\alpha}\, L_2^{\beta}\, L_3^{\gamma}\, dA^{(e)} = \frac{\alpha!\,\beta!\,\gamma!}{(\alpha+\beta+\gamma+2)!} \quad 2\Delta$$

A summation of values derived using this formula is presented in Table 4.1 for $(\alpha+\beta+\gamma) \leq 4$.

$$\frac{1}{\Delta}\int_{A^{(e)}} L_1^{\alpha}\, L_2^{\beta}\, L_3^{\gamma}\, dA^{(e)} = \frac{A}{B}$$

$\alpha+\beta+\gamma$	α	β	γ	A	B
0	0	0	0	1	1
1	1	0	0	1	3
2	2	0	0	2	12
2	1	1	0	1	12
3	3	0	0	6	60
3	2	1	0	2	60
3	1	1	1	1	60
4	4	0	0	12	180
4	3	1	0	3	180
4	2	2	0	2	180
4	2	1	1	1	180

Table 4.1

With this preliminary work finished, the actual derivation of the element matrices may now begin. In all, five matrices will be completely evaluated while one matrix, $\left[K_1\right]$, will have only two of its terms derived, due to the extensive amount of time and paper needed to evaluate $\left[K_1\right]$ in total. Beginning with this above-mentioned matrix as it appeared in Subsection A,

$$K_1(i,j) = \nu \int_{\Omega^{(e)}} \left(\frac{\partial Ni}{\partial x}\frac{\partial Nj}{\partial x} + \frac{\partial Ni}{\partial y}\frac{\partial Nj}{\partial y}\right) dxdy \qquad (4.20)$$

where $\Omega^{(e)}$ is the elemental area representing the solution domain, and i and j both vary from one to six. Since $\left[K_1\right]$ is an array multiplying the nodal variables of velocity and temperature, it must be correlated with the quadratic interpolation functions of equation 4.19. For the point (1,1), equation 4.20 becomes

$$K_1(1,1) = \nu \int_{\Omega^{(e)}} \left(\frac{\partial N_1}{\partial x}\frac{\partial N_1}{\partial x} + \frac{\partial N_1}{\partial y}\frac{\partial N_1}{\partial y}\right) dxdy \qquad (4.21)$$

where

$$\frac{\partial N_1}{\partial x} = \frac{\partial L_1}{\partial x}(4L_1-1) = \frac{b_1}{2\Delta}(4L_1-1)$$

and

$$\frac{\partial N_1}{\partial y} = \frac{\partial L_1}{\partial y}(4L_1-1) = \frac{C_1}{2\Delta}(4L_1-1)$$

substituting these values into equation 4.21

$$K_1(1,1) = \nu \int_{\Omega^{(e)}} \left[\frac{b_1^{\,2}}{4\Delta^2}(4L_1-1)^2 + \frac{c_1^{\,2}}{4\Delta^2}(4L_1-1)^2 \right] \; dxdy$$

$$= \frac{\nu(b_1^{\,2}+c_1^{\,2})}{4\Delta^2} \int_{\Omega^{(e)}} (16L_1^{\,2} - 8L_1 + 1) \; dxdy$$

employing Table 4.1 for these three cases above in which $(\alpha+\beta+\gamma) = 2, 1,$ and 0 respectively; plus the relationship that $\int_{\Omega^{(e)}} dxdy = \Delta$

$$K_1(1,1) = \frac{\nu(b_1^{\,2}+c_1^{\,2})}{4\Delta^2}(16 \cdot \frac{2}{12} - 8 \cdot \frac{1}{3} + 1)\,\Delta$$

or finally

$$\boxed{K_1(1,1) = \frac{\nu(b_1^{\,2}+c_1^{\,2})}{4\Delta}}$$

Next, consider the point (2,4) where

$$K_1(2,4) = \nu \int_{\Omega^{(e)}} \left(\frac{\partial N_2}{\partial x}\frac{\partial N_4}{\partial x} + \frac{\partial N_2}{\partial y}\frac{\partial N_4}{\partial y} \right) dxdy \qquad (4.22)$$

where

$$\frac{\partial N_2}{\partial x} = 4 \left[L_1\left(\frac{b_2}{2\Delta}\right) + L_2\left(\frac{b_1}{2\Delta}\right) \right],$$

$$\frac{\partial N_2}{\partial y} = 4 \left[L_1\left(\frac{c_2}{2\Delta}\right) + L_2\left(\frac{c_1}{2\Delta}\right) \right],$$

$$\frac{\partial N_4}{\partial x} = 4 \left[L_2\left(\frac{b_3}{2\Delta}\right) + L_3\left(\frac{b_2}{2\Delta}\right) \right],$$

$$\frac{\partial N_4}{\partial y} = 4 \left[L_2\left(\frac{c_3}{2\Delta}\right) + L_3\left(\frac{c_2}{2\Delta}\right) \right]$$

substituting these four values into equation 4.22

$$K_1(2,4)=\nu\int_{\Omega^{(e)}}\left\{16\left[L_1L_2\left(\frac{b_2}{2\Delta}\right)\left(\frac{b_3}{2\Delta}\right)+L_2^{\,2}\left(\frac{b_1}{2\Delta}\right)\left(\frac{b_3}{2\Delta}\right)\right.\right.$$
$$+L_1L_3\left(\frac{b_2}{2\Delta}\right)^2+L_2L_3\left(\frac{b_1}{2\Delta}\right)\left(\frac{b_2}{2\Delta}\right)+L_1L_2\left(\frac{c_2}{2\Delta}\right)\left(\frac{c_3}{2\Delta}\right)$$
$$\left.\left.+L_2^{\,2}\left(\frac{c_1}{2\Delta}\right)\left(\frac{c_3}{2\Delta}\right)+L_1L_3\left(\frac{c_2}{2\Delta}\right)^2+L_2L_3\left(\frac{c_1}{2\Delta}\right)\left(\frac{c_2}{2\Delta}\right)\right]\right\}dxdy$$

simplifying again, through the use of Table 4.1

$$\boxed{K_1(2,4)=\frac{\nu}{3\Delta}(b_2b_3+c_2c_3+2b_1b_3+2c_1c_3+b_1b_2+c_1c_2+b_2^{\,2}+c_2^{\,2})}$$

As can be seen, terms in the K_1 matrix can be quite lengthy and require considerable time to derive. On a more positive note though, this square matrix is symmetric and thus only half the terms need be calculated manually.

Next, attention will be focused on the K_2 and K_3 matrices. These two arrays may be considered together since the only difference between the two of them is that b values are associated with $\left[K_2\right]$ and c values with $\left[K_3\right]$. Otherwise, they are identical. These two matrices were given as

$$K_2(i,j) = \int_{\Omega^{(e)}}\left(\frac{\partial N_j}{\partial x}N_i^{\,P}\right)dxdy \tag{4.23}$$

and

$$K_3(i,j) = \int_{\Omega^{(e)}}\left(\frac{\partial N_j}{\partial y}N_i^{\,P}\right)dxdy \tag{4.24}$$

Taking the (3,6), equation 4.23 becomes

$$K_2(3,6) = \int_{\Omega^{(e)}} \left(\frac{\partial N_6}{\partial x} N_3^P\right) dxdy$$

where

$$\frac{\partial N_6}{\partial x} = 4\left[L_1\frac{\partial L_3}{\partial x} + L_3\frac{\partial L_1}{\partial x}\right]$$ and $N_3^P = L_3$, then

substituting above

$$K_2(3,6) = 4\int_{\Omega^{(e)}} \left[L_1\left(\frac{b_3}{2\Delta}\right)+L_3\left(\frac{b_1}{2\Delta}\right)\right] L_3 dxdy$$

$$= \frac{2}{\Delta}\int_{\Omega^{(e)}} (L_1L_3b_3+L_3^2b_1)\, dxdy$$

once again, using Table 4.1

$$\boxed{K_2(3,6) = \frac{1}{6}(2b_1+b_3)}$$

and consequently

$$\boxed{K_3(3,6) = \frac{1}{6}(2c_1+c_3)}$$

Following the same procedure throughout each of these 3x6 matrices, complete $\left[K_2\right]$ and $\left[K_3\right]$ are

$$\left[K_2\right] = \frac{1}{6}\begin{bmatrix} b_1 & b_1+2b_2 & 0 & b_2+b_3 & 0 & b_1+2b_3 \\ 0 & 2b_1+b_2 & b2 & b_2+2b_3 & 0 & b_3+b_1 \\ 0 & b_1+b_2 & 0 & 2b_2+b_3 & b_3 & 2b_1+b_3 \end{bmatrix}$$

and also

$$\left[K_3\right] = \frac{1}{6}\begin{bmatrix} c_1 & c_1+2c_2 & 0 & c_2+c_3 & 0 & c_1+2c_3 \\ 0 & 2c_1+c_2 & c_2 & c_2+2c_3 & 0 & c_3+c_1 \\ 0 & c_1+c_2 & 0 & 2c_2+c_3 & c_3 & 2c_1+c_3 \end{bmatrix}$$

The next two matrices to be derived, that is $\left[K_2\right]^T$ and $\left[K_3\right]^T$, can simply be written down by inspection of the two above arrays. Thus

$$\left[K_2\right]^T = \frac{1}{6}\begin{bmatrix} b_1 & 0 & 0 \\ b_1+2b_2 & 2b_1+b_2 & b_1+b_2 \\ 0 & b_2 & 0 \\ b_2+b_3 & b_2+2b_3 & 2b_2+b_3 \\ 0 & 0 & b_3 \\ b_1+2b_3 & b_3+b_1 & 2b_1+b_3 \end{bmatrix}$$

while its counterpart is then

$$\left[K_3\right]^T = \frac{1}{6}\begin{bmatrix} c_1 & 0 & 0 \\ c_1+2c_2 & 2c_1+c_2 & c_1+c_2 \\ 0 & c_2 & 0 \\ c_2+c_3 & c_2+2c_3 & 2c_2+c_3 \\ 0 & 0 & c_3 \\ c_1+2c_3 & c_3+c_1 & 2c_1+c_3 \end{bmatrix}$$

Finally, the last elemental matrix to be analyzed is the one associated with the time-dependent nodal parameters. In subsection A this matrix was given as $\left[CD\right]$. For convenience here, let $\left[CD\right] = \left[K_t\right]$, then

$$K_t(i,j) = \int_{\Omega^{(e)}} N_i N_j \, dxdy \tag{4.25}$$

with both i and j running from one to six. Consider, for example, point (1,5)

$$K_t(1,5) = \int_{\Omega^{(e)}} N_1 N_5 \, dxdy$$

substituting from equation 4.19

$$K_t(1,5) = \int_{\Omega^{(e)}} (2L_1^2-L_1)(2L_3^2-L_3)\,dxdy$$

$$= \int_{\Omega^{(e)}} (4L_1^2L_3^2-2L_1L_3^2-2L_1^2L_3+L_1L_3)\,dxdy$$

then from Table 4.1, using $(\alpha+\beta+\gamma)$ four separate times, this term reduces rather easily to

$$\boxed{K_t(1,5) = -\frac{\Delta}{180}}$$

Factoring out a constant of $\frac{\Delta}{180}$, the total K_t matrix takes on the form

$$\left[K_t\right] = \left[CD\right] = \frac{\Delta}{180}\begin{bmatrix} 6 & 0 & -1 & -4 & -1 & 0 \\ 0 & 32 & 0 & 16 & -4 & 16 \\ -1 & 0 & 6 & 0 & -1 & -4 \\ -4 & 16 & 0 & 32 & 0 & 16 \\ -1 & -4 & -1 & 0 & 6 & 0 \\ 0 & 16 & -4 & 16 & 0 & 32 \end{bmatrix}$$

Which is also a symmetric matrix, thereby allowing faster derivation of the individual terms with less chance of numerical error.

C. STRUCTURE OF COMPUTER PROGRAMS FOR FLOW ANALYSIS

A total of three computer programs analyzing two distinct test cases of fluid flow problems were employed in this thesis. The first was a steady state analysis of Couette flow. This involved determining the solution of the velocity profiles (linear and nonlinear) in a shear- and pressure-induced flow between flat parallel plates.

The upper plate slides in the positive x-direction with a constant velocity u, while the lower plate remains stationary. There is no velocity component normal to the plates, that is, v=0 in the y-direction. The governing equations for this particular fluid flow are

Continuity:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \qquad (4.26)$$

Momentum:

$$u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{1}{\rho}\frac{\partial P}{\partial x} + \nu\nabla^2 u \qquad (4.27)$$

$$u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{1}{\rho}\frac{\partial P}{\partial y} + \nu\nabla^2 v \qquad (4.28)$$

For determination of the velocity profile involving only linear terms, the left hand side of equations 4.27 and 4.28 are set equal to zero (no inertia terms).

A physical representation of the Couette flow analyzed is shown in Figure 5. Node and element numbering is the same as in Figure 2. A pressure gradient of -3 units is directed along the x-axis, i.e., $\frac{dP}{dx} = -3$.

The two remaining programs formed the majority of the theoretical portion of this study. They were devised to carry out the calculations for the analysis of two-dimensional or axisymmetric natural convection heat transfer problems.

The first is the lesser-complex steady state approach, whereby all transient conditions characteristic of the system are assumed to have died out, leaving only those steady state parameters remaining to be solved for. Once the finite element matrix equations describing the system are correctly assembled, a library subroutine (LEQT2F) functioning as a linear equation solver is called and the desired nodal parameters can be calculated. The second program takes into account the previously-neglected time dependence of the system by introducing a type of finite difference integration scheme to solve the transient portion of the governing equations. This integration technique must be an iterative procedure in order to circumvent the problem of nonlinearity similar to that resulting from the addition of inertia terms. Furthermore, the integral is solved at successive time steps, with time being increased until the value of the field variable converges, within tolerance, to that of its steady state counterpart. This segment of the total equation is then algebraically combined with the remaining steady state solution to yield values of nodal field variables with improved accuracy. Gravity acting in the longitudinal direction was taken into account for both the two-dimensional and axisymmetric flows.

The flow region to be studied is first defined, followed by the setting up of coordinate axes (Figure 1). The location of the origin of these axes is in most cases arbitrary, except that for a problem involving axial symmetry the x-axis must coincide with the system axis of symmetry. The flow region is then divided into a mesh of triangular elements, and the nodal points are numbered in the sequence previously described. Once the setting-up of the solution domain is completed, computer analysis of the system with its included boundary conditions can be initiated.

Structure of the steady state fluid mechanics problem will be discussed in detail, primarily because it comprises one entire program and with the addition of the transient stiffness matrix elements, accounts for the main program in the time-dependent study.

The program was coded in FORTRAN IV language and begins with a series of DIMENSION statements, which set up the arrays needed in the calculations. As indicated in statements 0260-0310, storage has been allocated for problems with up to 117 nodes; however larger problems can be considered by simply increasing the dimensions of these matrices. The limit of problem size is dictated by the

core storage of the available machine. Next, the program calls for a declaration of the type of program to be solved (either two-dimensional or axisymmetric); then the appropriate problem label is printed (statements 0360-0410). Before proceeding to input the data describing the finite element mesh, all the matrix arrays must be initialized by setting all terms in these arrays equal to zero.

Statements 0930-1150 read into the program the node numbers and the coordinates of the nodes for the complete finite element mesh. Also, the system topology, the element numbers, and the numbers of the six nodes associated with each element are read. Beginning with statement 1300, the velocity, pressure and temperature conditions within the solution domain are inserted. Correspondingly, conditions are specified for the QX, QY, QZC and QZ indices where the nodal field variables are unknown. Since the solution obtained by the program depends intimately on the body of data, the program is queried to print out all data that have been input. This enables the programmer to check for input data errors. Statement 2970 marks the completion of these steps; the program is now ready to commence work on a particular fluid mechanics problem.

The loop to begin calculating the various element matrices starts with statement 3010. Once the element matrices $\left[\text{TM\$}\right]$ are computed for one element, they are assembled into the master or system stiffness matrix $\left[\text{TM}\right]$ by the code followed in statements 7740-7790. The non-linear terms appearing in the velocity and temperature expressions of the governing equations are formulated in statements 4110-5900. An iterative process compares each of these terms with a corresponding quantity in the linear symmetric $\left[\text{TM\$}\right]$ matrix until they converge in value. It is this comparison value that is then assembled with all similar values of the other elements to form the system $\left[\text{TM}\right]$ matrix, which now exhibits or reflects the non-linearity.

Since each element in the triangular mesh has six (6) nodes, the local node numbers are I\$ = 1, 2, ...,6. The global node numbers for the element are recovered from the parameter NODE (K,I\$), which was read as input data for the element; that is, for element K, the node numbers N(1)=NODE(K,1), N(2)=NODE(K,2), etc. were introduced. Then the code in statements 7740-7790 loads the terms of the elemental matrices into their proper locations in the system matrices. Each time that a term of an element

matrix is placed in a location in the system matrix where another term has already been inserted, this new term is added to whatever value is there. A similar loading process takes place for the right-hand-side column vector in statements 7810-7940.

After all the elements have been processed in this fashion, the assembled system equations are ready to be modified to account for the boundary conditions or phenomena. This is done by statements 7980-8060. Thus, at the conclusion of statement 8060, the system equations possess the form

$$\begin{bmatrix} \mathrm{TM} \end{bmatrix} \begin{Bmatrix} \mathrm{X} \end{Bmatrix} = \begin{Bmatrix} \mathrm{RHS} \end{Bmatrix}$$

where $\begin{Bmatrix} \mathrm{RHS} \end{Bmatrix} = \begin{Bmatrix} \mathrm{Kq} \end{Bmatrix} = \begin{Bmatrix} \mathrm{QX+gBTm} \\ \mathrm{QY} \\ \mathrm{QZC} \\ \mathrm{QZ} \end{Bmatrix}$, and $\begin{Bmatrix} \mathrm{x} \end{Bmatrix} = \begin{Bmatrix} \mathrm{u} \\ \mathrm{v} \\ \mathrm{P} \\ \mathrm{T} \end{Bmatrix}$

Not all of the components of the column vector $\begin{Bmatrix} \mathrm{RHS} \end{Bmatrix}$ are known because the Q values at nodes where velocity, pressure or temperature is specified are unknown; that is, at each node i, either u_i, v_i, or T_i is known on the one side, or Kq_i is known on the other. A similar relationship exists at the corner nodes for pressure and QZC values. The only Q's that can be specifically labeled as

heat fluxes are the QZ's, since they directly relate to temperature parameters within the system.

The only thing that remains to be done now is to call a compatible linear equation solver to produce the nodal variables sought. In this case, LEQT2F was chosen because of its speed and accuracy.

The following is a list of the symbols and descriptions utilized in coding the above program:

Symbol	Description
NCASE	interger which specifies the type of problem to be solved: NCASE=1, 2-D plane problem NCASE=2, axisymmetric problem
NN	number of nodes in solution domain
NNCN	number of corner nodes
NE	number of elements
XC(I),YC(I)	global coordinates of node I
NODE(J,I)	J=1,2,....NE; I=1,2,....6 node numbers associated with element J
NVS(I)	node number where velocity or temperature is specified
NPS(I)	node number where pressure is specified
VELU	specified nodal u velocity
VELV	specified nodal v velocity
PNP	specified nodal pressure

Symbol	Description
TNT	specified nodal temperature
NQS(I)	node number where a Q value is specified; QX and QY are specified only at internal nodes, while QZC and QZ may be specified at either external or internal nodes
QXNS	specified nodal value of QX
QYNS	specified nodal value of QY
QZCNS	specified nodal value of QZC
QZNS	specified nodal value of heat flux QZ
XC$(I),YC$(I)	local coordinates of node I
TM$	element stiffness matrix
TM	system stiffness matrix
DEL	area of a triangular element

The program output begins with a statement declaring the type of problem to be solved - either nonlinear two-dimensional or axisymmetrical. Next, all input data are printed and labeled for easy identification. To ensure the validity of the solution, the printed input data should be carefully checked against the intended input. A statement following these printed data identifies which nodal parameters are associated with which system nodes (remembering specifications of the finite element analysis called for a value of both velocities along with a temperature

at each node, while a pressure value could be defined only at the corner nodes). The complete continuum solution follows in the form of a numbered list, in which the integer appearing at the far left of this list designates the node number, or multiple of it in cases above I=35, and the figure on the right representing the value of the nodal variable in double precision.

D. NUMERICAL RESULTS

Complete numerical listings of the field variables for both the Couette flow problem and the steady state heat transfer problem are shown in the two computer program outputs.

The velocity profile for the linear Couette flow, i.e. node numbers 1-5, 6-10, etc., revealed that the finite-element method of analysis agreed with the exact solution of this shear-type flow out to the sixth decimal place. This is evident by the fact that all five of the FEM points lie exactly on the smooth curve depicting the exact solution in Figure 15.

The approximate steady state isotherms of Figure 16 are directly related to the nonlinear temperatures (node numbers 83-117 of the second set of nodal variables) in

the heat transfer problem. These isotherms, or constant nondimensional temperature lines, vary in value from +1.0 on the hot temperature wall, to -1.0 on the cold temperature wall. The equation used for deriving these values at all thirty-five nodal points within the solution domain was

$$\theta = \frac{T-T_M}{T_H-T_M} \tag{4.29}$$

where T is the nodal temperature and T_M is the mean temperature of the fluid defined at $T_M = \frac{(T_H+T_C)}{2}$.

The general shape and relative location of the various isotherms within the rectangular enclosure are somewhat similar to those of comparable heat transfer flows involving different Prandtl numbers, Grashof numbers and L/D ratios. However, due to the relatively low Grashof number of the present system (946.4), there is a total lack of a plateau in the center region of Figure 16 and the closely packed boundary layer flow near the walls is also missing. This boundary layer type flow is characteristic of much higher Grashof numbers such as those found in the comparative examples in [3], [7] and [15] where the Gr_L ranged from 5000 up to 18000.

Based solely on the thirty-five nodal point temperatures available from the solution, the isotherms were sketched as

shown in Figure 16. Lacking additional information, the shape of the contour lines between such nodes were linearly approximated, to a large extent, without speculating as to their exact curvature.

The actual height and width of the enclosure was normalized to y^* and x^*, respectively, for easier interpretation of the figure.

V. EXPERIMENTAL PROCEDURE

A. ARRANGEMENT OF TEST APPARATUS

The experimental apparatus was arranged so as to allow the study of an essentially two-dimensional fluid flow.

The major components of the apparatus consisted of the test platform, which housed the rectangular enclosure (Figure 6), a control system made up of two water circulators that maintained the vertical copper walls of the test platform at desired temperatures (Figure 7), and a large (250 mm DIA) plano-convex glass lens for reducing the object (8.5 x 1.875 inch vertical rectangular cavity) down to a smaller image size that could be completely captured on the 4x5 inch holographic plate (Figure 11).

The rectangular enclosure holding the fluid under investigation was 8.5 inches high, 7 inches long, and 1.875 inches wide. This test cavity was sandwiched between two plexiglas water reservoirs providing constant circulation by means of manifold connections on their tops and bottoms. Hot and cold water drains were located on top of the left and right reservoirs, respectively. Similarly, on the bottom were the hot and cold water inputs. The

inner walls of these two reservoirs were formed by quarter inch thick oxygen-free copper plates. Also, these same copper plates comprised the principle walls of the rectangular enclosure, with the "side walls" being made of plate glass, in order to allow visual observations. Six thermocouples were attached to each copper wall and then connected to a multichannel recorder for temperature monitoring purposes. The time needed for each copper plate to reach its respective equilibrium temperature once the water circulators had been turned on was approximately 39.8 seconds. For comparison purposes, it took the system just under one hour (58.1 minutes) for the 50-HB-3520 lubricating fluid to attain a steady equilibrium temperature under the same experimental conditions.

An important constraint imposed by interferometry is that the total distance traveled by the object beam must be nearly identical with the total path length of the reference beam, if the index of refraction throughout is uniform. Since the fluid in the rectangular enclosure possessed a refraction index of 1.461 and laser light along the object beam had to travel through 7 inches of this fluid, then the corrected path length through the test cavity was 10.23 inches, or a net increase of over 3 inches.

With this in mind, the equipment was arranged in a semi-elliptical pattern on a heavy table supported at six critical areas by inflated inner tubes. These were to act as stabilizing devices. Equipment could not be arranged on an exact ellipse due to the fact that a distance of four feet, five inches alone was needed from the object to the plano-convex lens out of a total table length of eight feet. Even so, the turning mirrors were located on the apexes of the "shortened" minor axis, the beam splitter at one end of the major axis, and the aqueous hologram holder at the opposite end (Figure 8). Spatial filters were employed to clean up each beam and expand it. A diverging lens was inserted just after the object beam spatial filter in order to expand this beam to proper size before it reached the rectangular test slit. Also, a collimating lens was placed between the spatial filter and hologram holder along the reference beam. Finally, a large diffusing screen made from a piece of developed film mounted on plexiglas and secured in a rigid metal frame, plus the test platform itself, were placed between the object mirror and the hologram holder (Figures 9 and 10).

Choosing the correct hologram holder is very important in live fringe holography. The reconstructed virtual image

must exactly match the original object. Unfortunately, after the processing of a hologram, the emulsion on its surface tends to dry, thus causing an unwanted displacement of the virtual image. One procedure that may be used to circumvent this problem is to choose a holder that maintains the hologram in aqueous surroundings, such as was utilized in this experiment. Also, in order to keep the hologram perfectly rigid during and after processing, the exposed glass plate was secured in a removable metal frame complete with handle. In this way, exact replacement in the hologram holder after processing posed no problem. Two micrometers built into the holder's top and left side were then used for fine adjustment of the hologram.

The test cavity or rectangular enclosure was filled with a very highly viscous fluid (actually a lubricant) produced by Union Carbide and known as "UCON" 50-HB-3520. This fluid was required to possess physical properties such that Rayleigh numbers, based on cavity width, of the order of 10^4-10^5 could be obtained in the apparatus, at accurately measurable temperature differences. Since (T_H-T_C) was held constant throughout the experiment, only one Rayleigh number was calculated. Its value of 1.0755×10^4 was well within the above tolerance zone. Worth mentioning

is the generally accepted prediction that above $Ra \simeq 2.0 \times 10^4$, the phenomena known as "secondary flows" begin to occur. Obviously, such was not the case in this experiment.

A scribed grid pattern was attached to the back side of the test cavity to assist in alignment of the fringes. Two water circulators were connected by tubes to manifold nipples on the reservoir ends of the test platform. One circulator was set to deliver distilled water at 20°C (cold temp.) and the other at 25°C (hot temp). By using slide valves, the amount of water expended from the circulators could be regulated and controlled.

The experimental procedure was initiated only after the entire system had been carefully aligned. The rectangular enclosure was allowed to sit undisturbed for a period of at least several hours to ensure an equilibrium temperature state throughout the fluid. Then, a shutter was placed directly in front of the beam of a 3 milliwatt, helium-neon continuous wave laser serving as the coherent light source. After an Agfa-Gevaert Inc. 10E75 holographic recording plate was placed in the holder, the shutter was opened and the plate was exposed for one and one-half seconds (Figure 13). This plate was then removed, developed, and replaced in its exact position. Both circulators were

then turned on, and a continuous flow of water at 20°C and 25°C was allowed to cycle through the reservoirs on the test platform. Once fringe lines appear, their visibility can be strengthened by following the procedure outlined in the next subsection.

If one word could be used to describe the single most important factor determining the success or failure of this experiment, it would have to be rigidity. All relatively light-weight gear, such as; the laser, turning mirrors, spatial filters, and the plano-convex lens had to be weighted down to make them immovable. The test platform, in which the rectangular cavity was located, was sufficiently heavy on its own to preclude it from having to be additionally weighted down. The hologram holder already came with a very heavy base attached. All connecting devices, including the tubes transporting heated water to the plexiglas reservoirs and plastic sleeves housing the thermocouple leads were securely taped together to prevent vibration or motion. Any such random vibration would cause the fringe patterns to become lost.

The viewing of these fringes and the subsequent collecting of data can be accomplished by positioning the appropriate camera in a direct line with the rectangular

enclosure, plano-convex lens, and hologram holder (Figure 11). A television monitor (Figure 12) was employed for convenient viewing of the fringe patterns in an area adjacent to the experimental set-up.

B. HOLOGRAPHIC INTERFEROMETRY APPLICATIONS

Holographic interferometry is an excellent technique for developing interference fringe patterns, which may in turn be evaluated to quantitatively provide an accurate temperature field throughout the domain of the system.

Heat convection in a rectangular cavity would be difficult to analyze empirically. However, by replacing the sensors that would ordinarily be used to record temperature changes and flow rates with holographic technology, one can analyze directly the variation of the density fields within the rectangular enclosure. This technique also eliminates the inherent change in temperature and flow pattern caused by the physical insertion of the sensors into the test fluid.

Real-time holographic interferometry allows a continuous flow of information to be recorded at the precise time any changes in the observed fluid occur. Single exposure holograms are utilized with real-time interferometry. A time

sequence can be derived for each different viewing position, with the use of a single developed hologram.

Such an exposure technique consists of recording phase and amplitude information from an object, in this case the rectangular fluid enclosure, onto a holographic plate. The recording is accomplished through the use of a reference and an object (scene) beam, originating from a single source (Figure 13). After processing, the hologram is accurately repositioned in its holder. Illuminating the plate with the original reference beam results in the primary (virtual) image being projected onto the same area as was the object (Figure 14). By focusing the object and virtual image beams onto a film or focal plane, and then adjusting the system so that the two interfering wavefronts (object wave and reconstructed wave) coincide, fringes can be produced.

The hologram now can be finely adjusted to orient the fringes in either a vertical or horizontal reference frame. Likewise at this time, the fringe patterns can be made to appear more visible by varying the beamsplitter to increase the intensity of the reference beam while decreasing that of the scene beam. If the original object is changed or altered in any fashion by the effect of temperature, motion, or pressure, an exact superposition will create a reinforcement

or cancellation of the intensities of the two waves with the result being the establishment of a fringe pattern. A dark fringe is produced whenever the difference between the object and reconstructed wavefront involves an odd factor of $\pi/2$. Bright fringes occur when this difference equals an integer value of 2π.

By inserting a camera in-line with the scene (object) beam, but on the back side of the holographic plate, one can observe and record live fringe data. This technique provides a real time analysis of an unsteady system without the need for expensive and time consuming sensors and calibration.

After processing has been completed, problems arising from live fringe single exposure holography include; displacement of the virtual image due to drying emulsions on the holographic plate, and non-exact replacement of the hologram in its holder. If any relative motion whatsoever has transpired between pieces of the experimental equipment during or after replacement of the hologram, the fringe patterns may be destroyed.

It was this last problem that caused a particularly detrimental effect on the experimental results of this thesis. Somewhere within the system (apparatus arrangement)

there existed a source of motion or a piece of gear slightly off the horizontal reference plane that completely eliminated these fringe formations almost immediately after they evolved. The exact source(s) was never totally isolated, but the possible choices were reduced down to two, the water circulators and/or the support stand of the hologram holder.

VI. CONCLUSIONS

The finite-element method was incorporated into steady state and time dependent computer programs for analyzing laminar convective heat transfer between parallel plates. Two sample cases were tested utilizing the general steady state program. In each case, values of derived field variables compared favorably to either an exact solution, in the case of the Couette flow problem (Figure 15); or to similar theoretical results, in the case of the heat transfer problem (Figure 16). The exact solution of the velocity profile for Couette flow was obtained by programming the analytical expression given by equation 5.5 in [11].

After successful steady state results were achieved, a second computer program was then designed to take into account the previously-neglected transient behavior of the system. A major portion of the steady state fluid mechanics problem was interfaced with a series of subroutines, wherein the time-dependent terms were to be calculated, to yield a total solution to the governing system of equations and associated boundary conditions. Time itself became a limiting factor in the completion of this second program.

A problem associated with the convergence of the field variables remains to be resolved.

In the experimental phase of this thesis, five attempts were made to produce live fringe formations through the use of holographic interferometry. In only one of these attempts was there observed a <u>momentary</u> interference pattern, corresponding to the temperature gradient, across the test section. This observation lasted approximately three (3) seconds after the water heaters/circulators were activated.

The main factor(s) influencing this inability to acquire such live fringes, on film, was the necessary exclusion of the hologram holder from the recording plane because of limited table length and/or the vibrations generated by the water circulators used in the experiment. Either of these detrimental conditions could have eliminated completely the formation of interference fringe patterns.

Due to considerable time delay in the acquisition of some of the experimental apparatus, no further documentation of real time holographic interferometry study could be made beyond the previously-mentioned five attempts.

APPENDIX A

FIGURES

Figure 1
Rectangular Enclosure

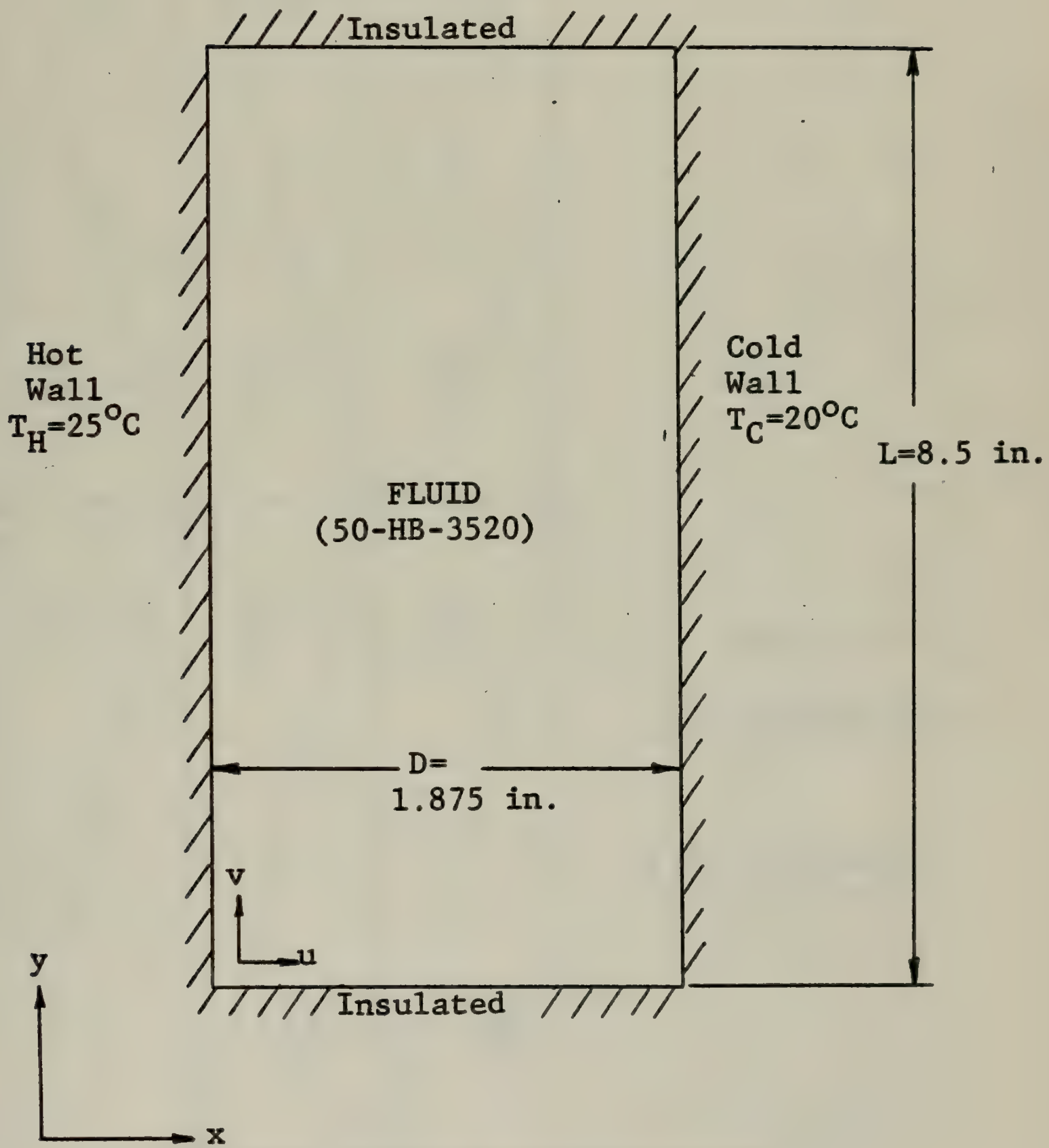


Figure 2
Discretization of Solution Domain

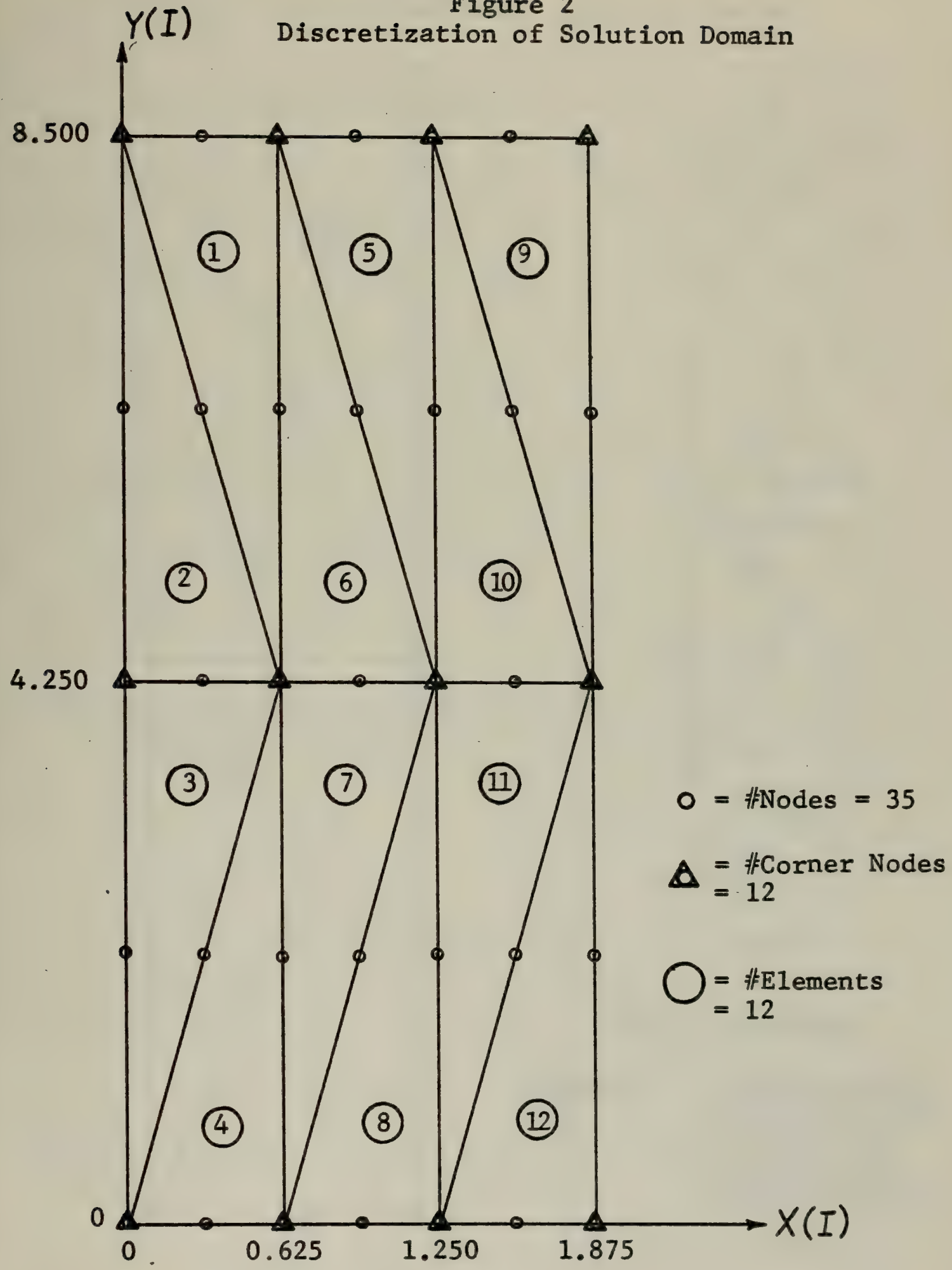
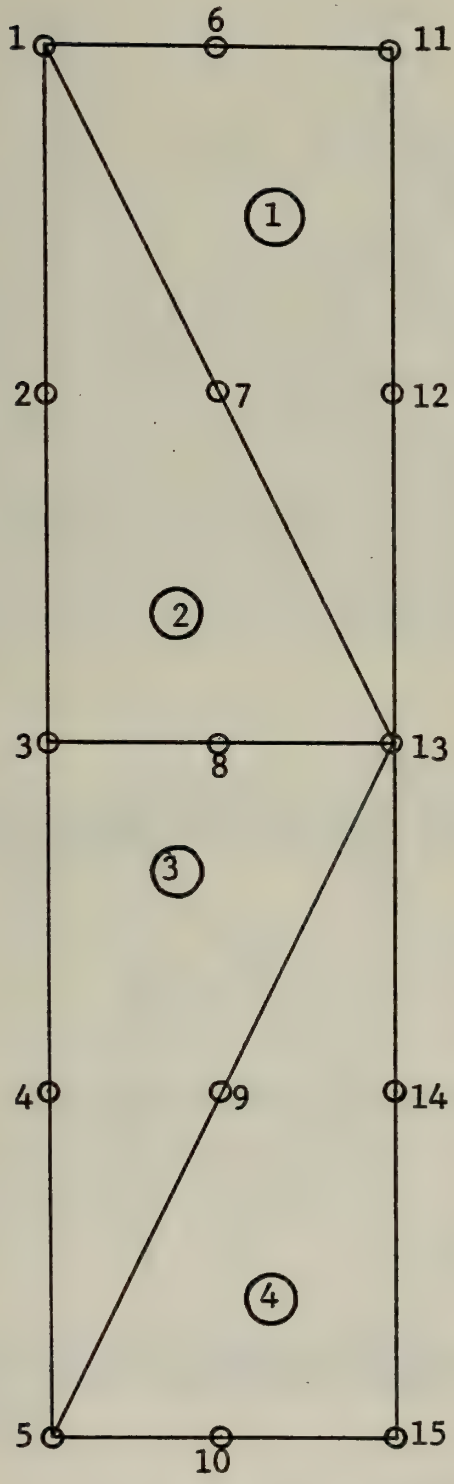
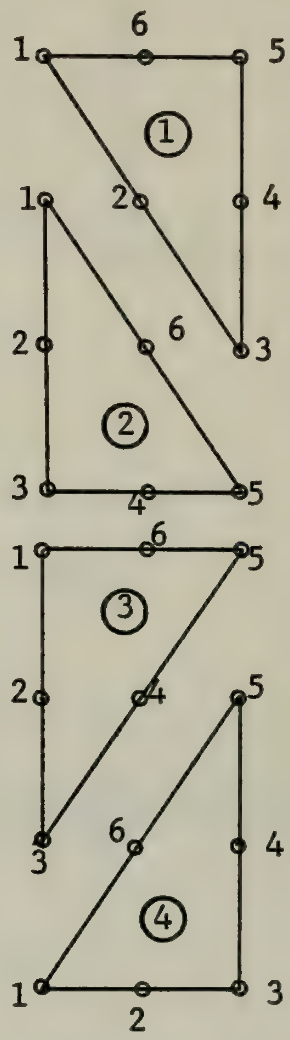


Figure 3
Global Node-Numbering vs. Local Numbering



Global



Local

System Topology

Element#	Global Node#'s
1	1,7,13,12,11,6
2	1,2,3,8,13,7
3	3,4,5,9,13,8
4	5,10,15,14,13,9

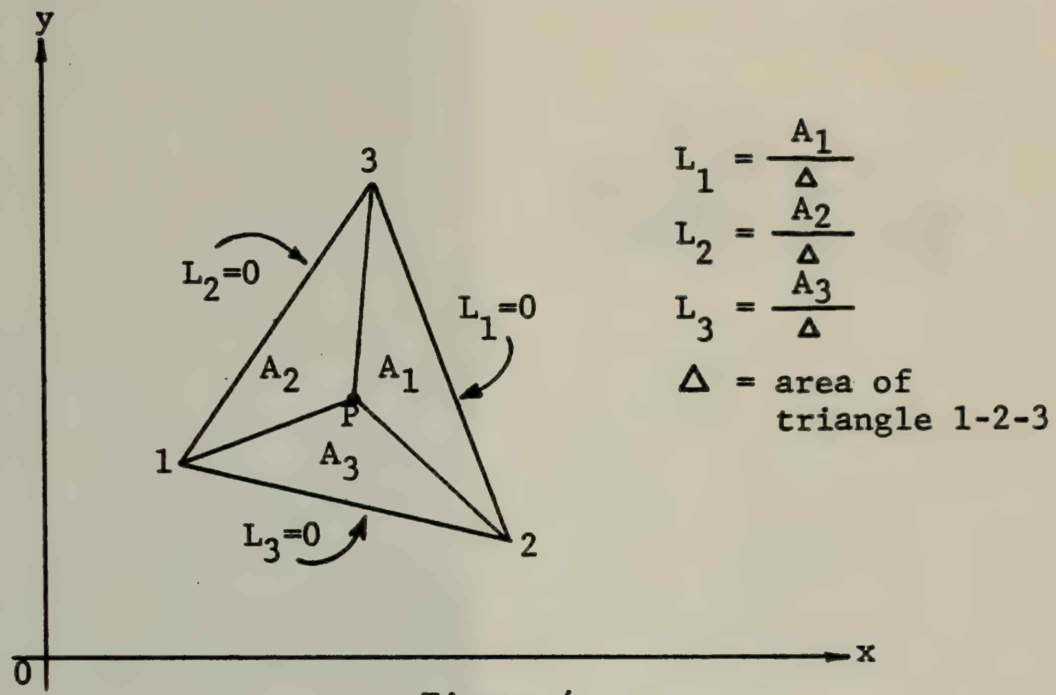


Figure 4
Area Coordinates for a Triangle

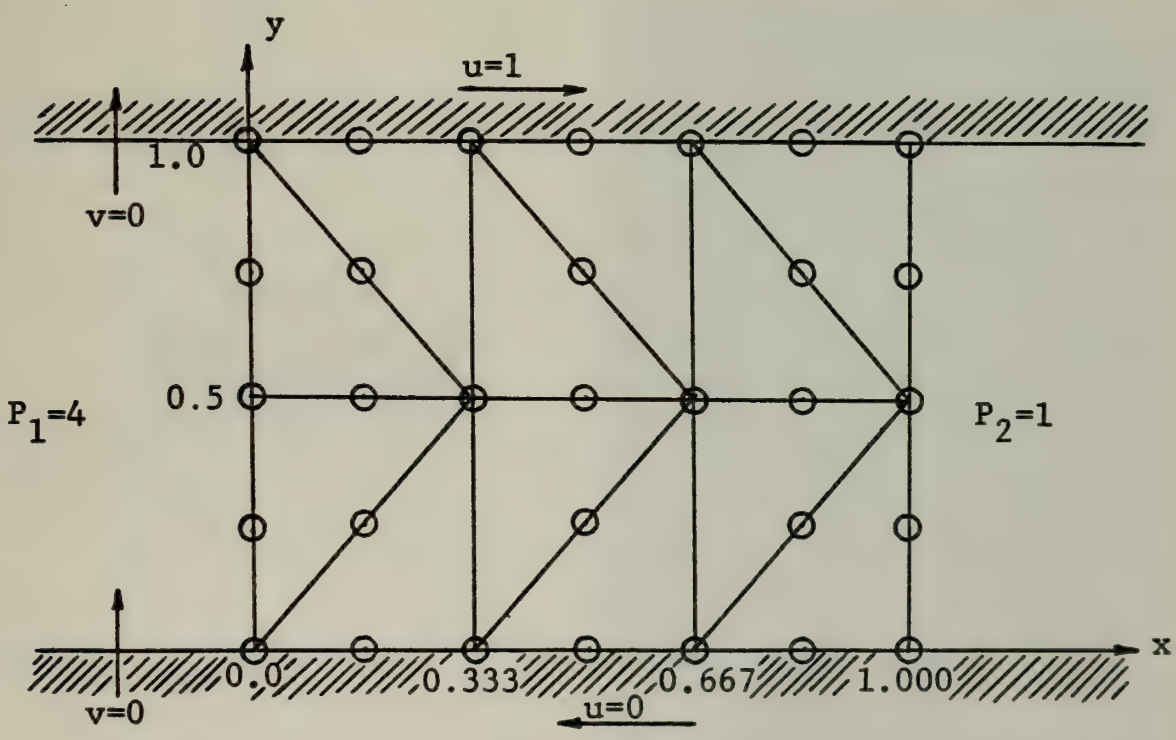


Figure 5
F.E.M. Analysis of Couette Flow

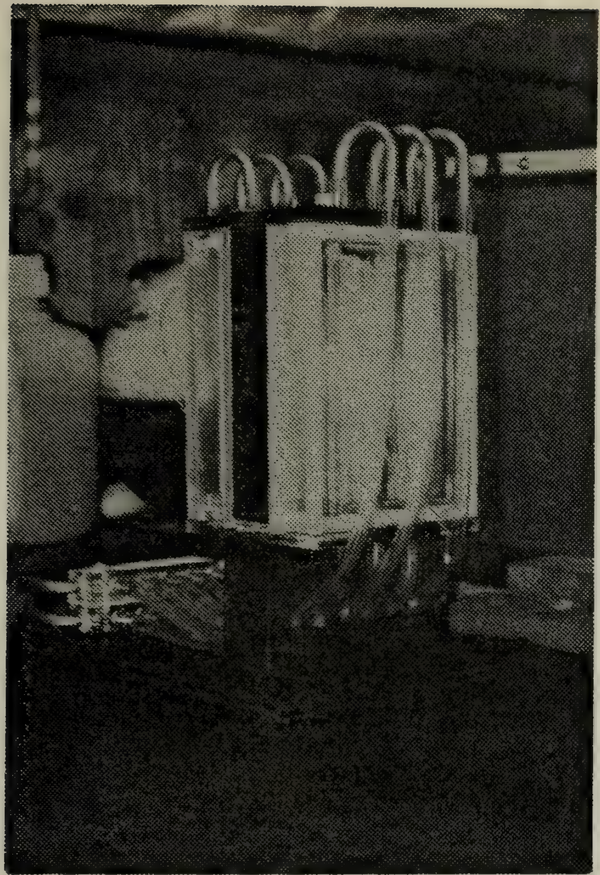


Figure 6
Test Platform with
Rectangular Enclosure
and Water Reservoirs

Figure 7
Water Heaters and
Circulators with
Connecting Tubes

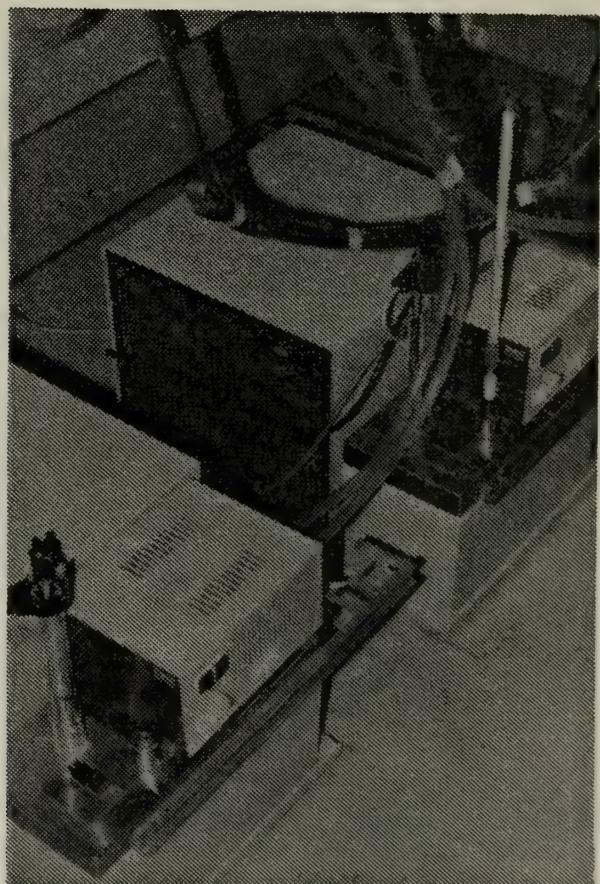
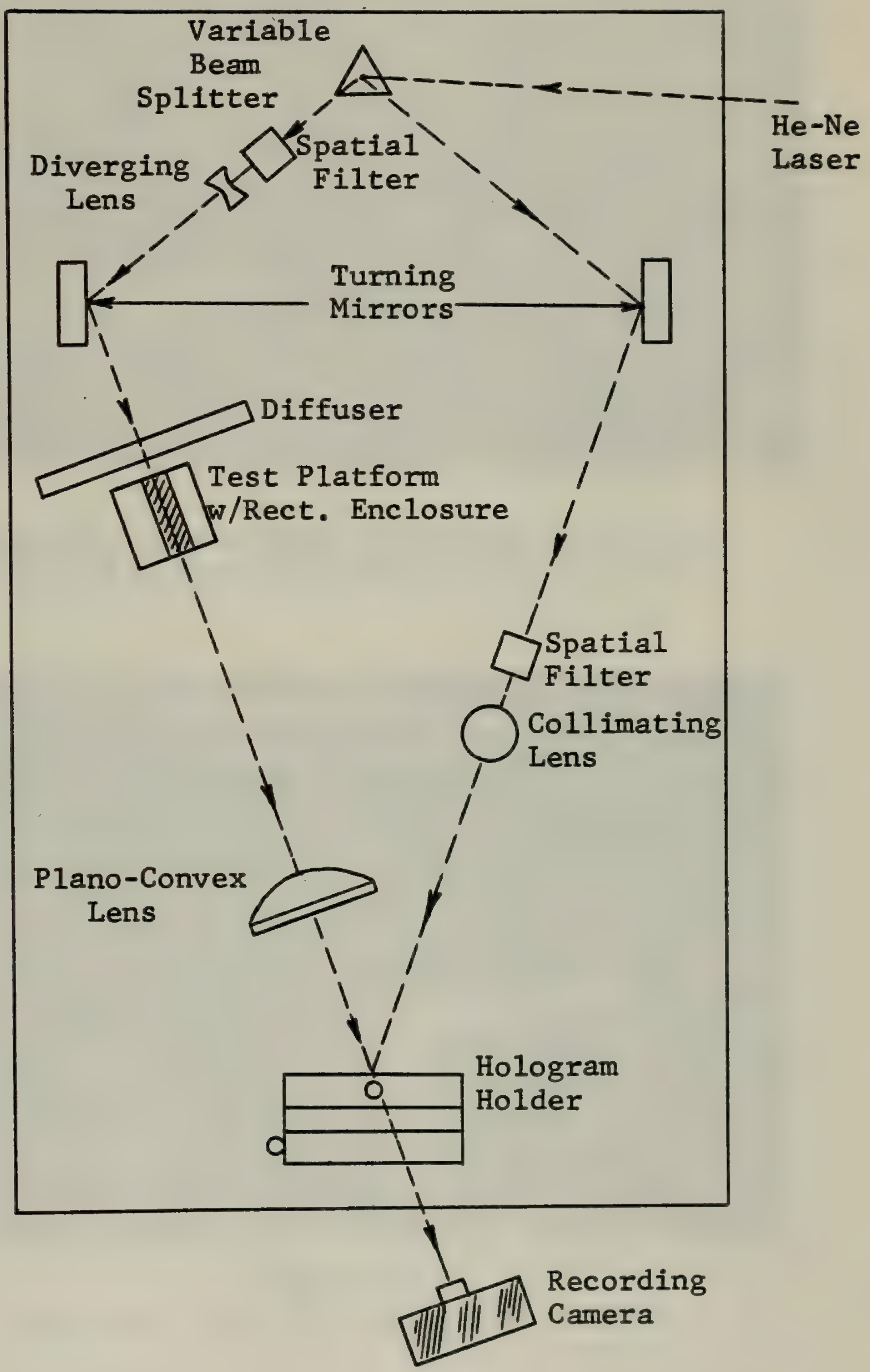


Figure 8
Table Arrangement (top view)



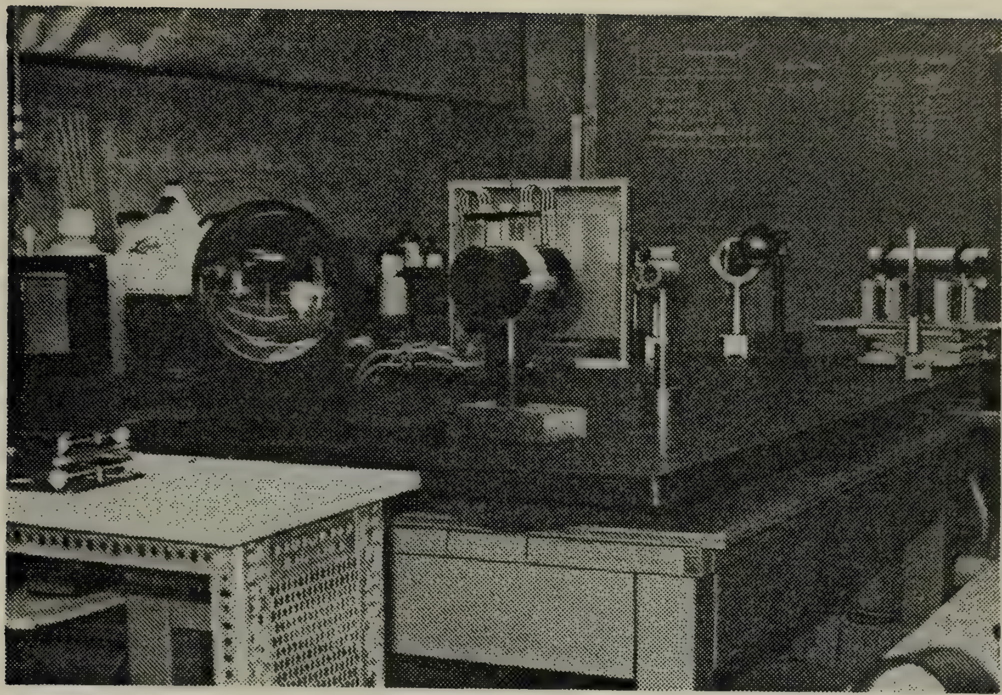


Figure 9
Apparatus Arrangement with Reference Beam
Oriented on the Right-hand-side of Table

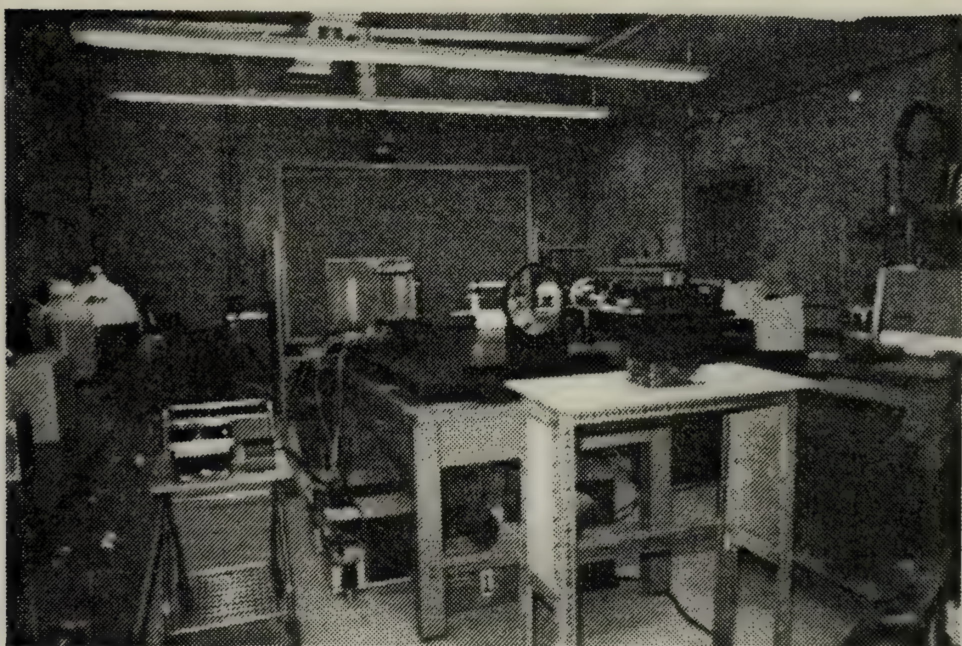


Figure 10
Panoramic View of Experimental Layout

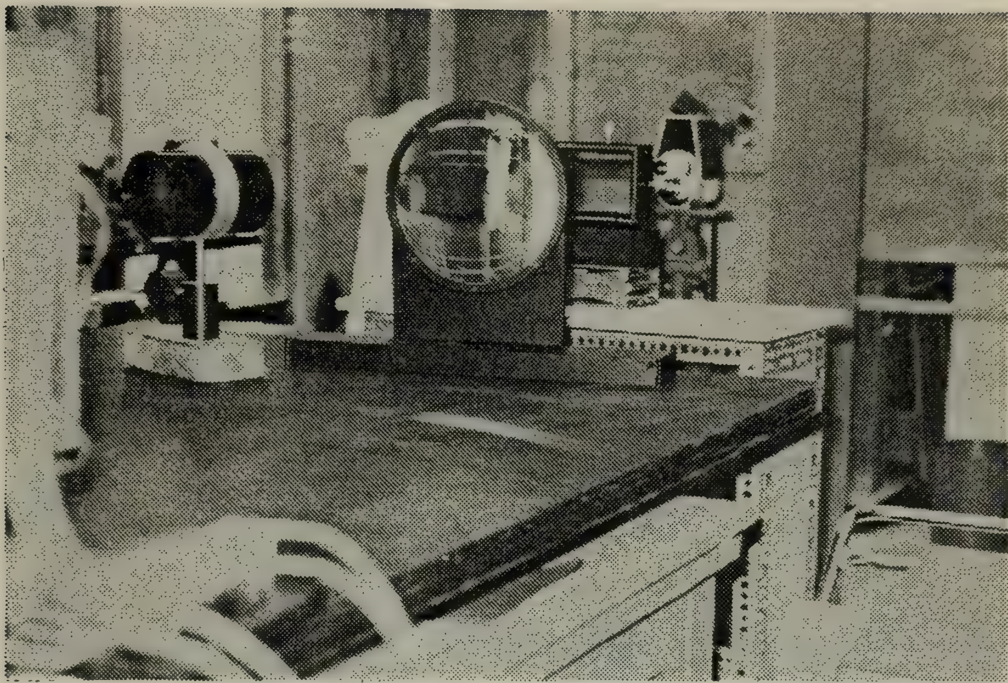


Figure 11
Direct Line-up on Object Beam from
Test Platform to Recording Camera

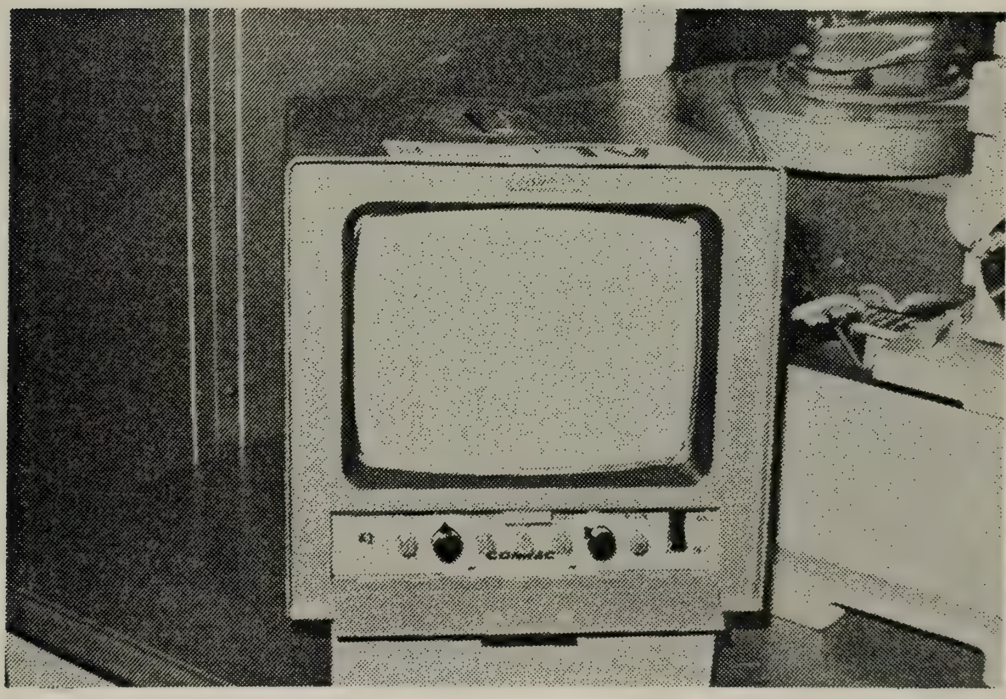


Figure 12
Television Monitor Used for Convenient Viewing of Fringes

Figure 13
Holographic Recording (top view)

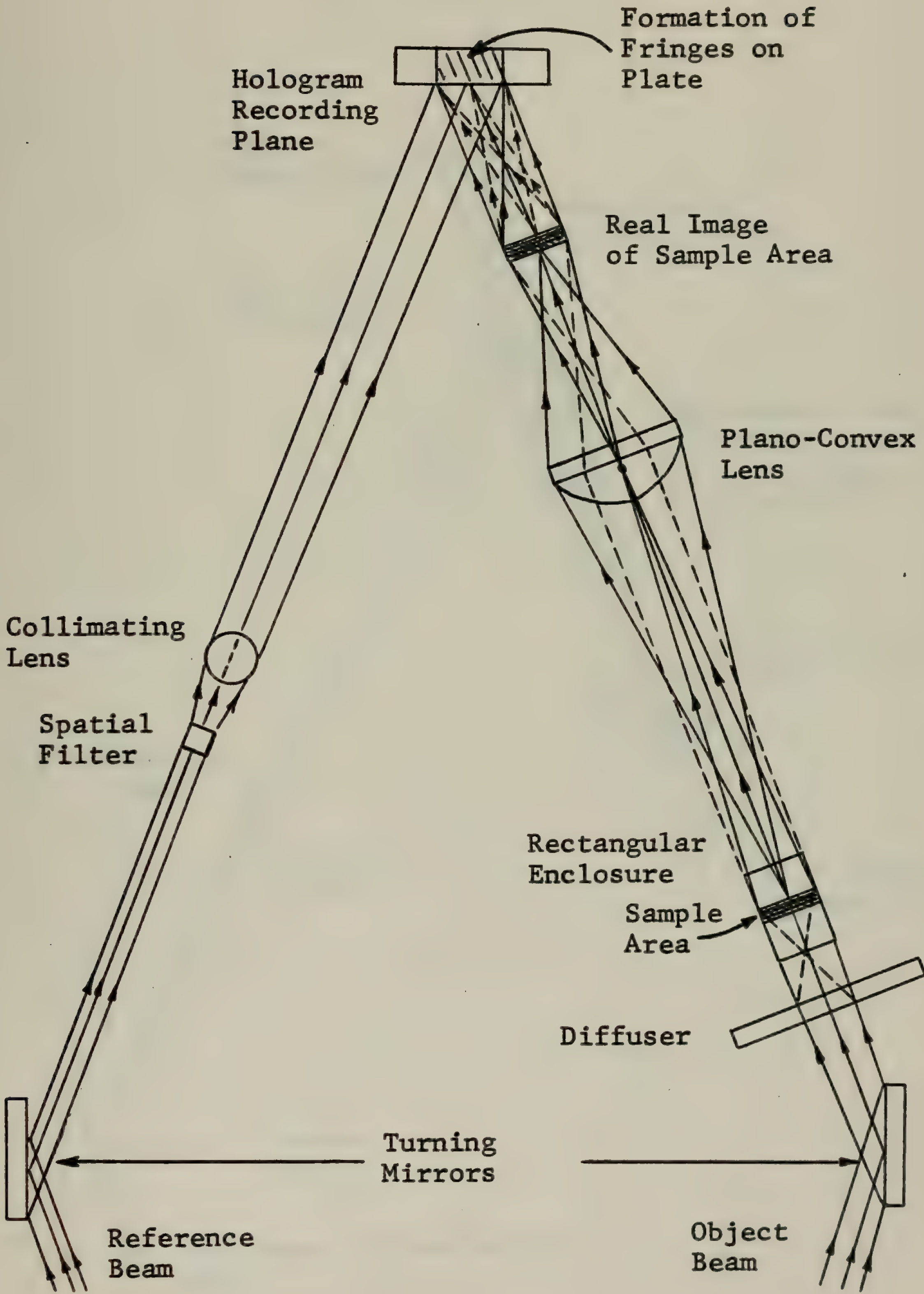
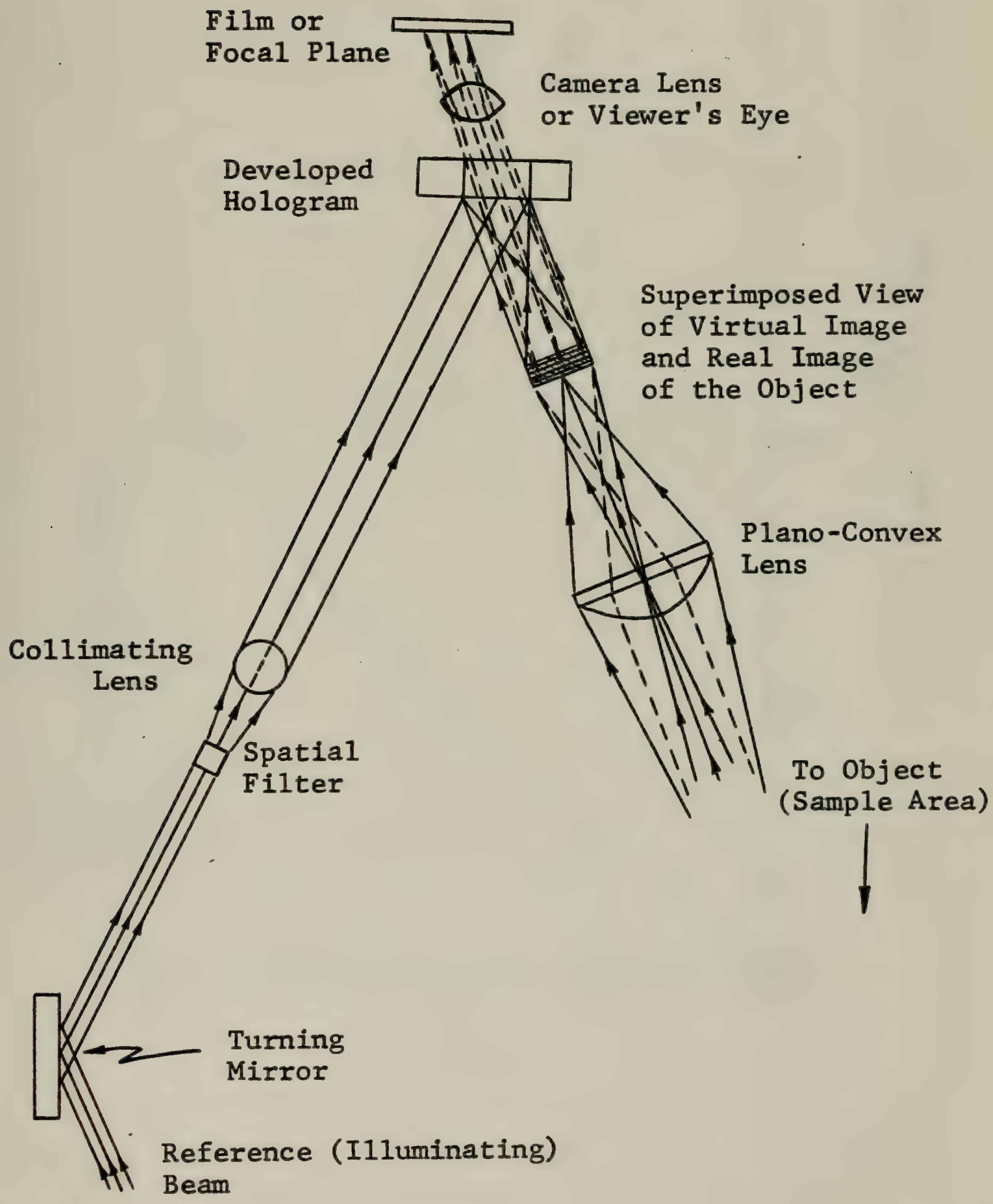


Figure 14
Reconstruction and Recording of
Interference Patterns



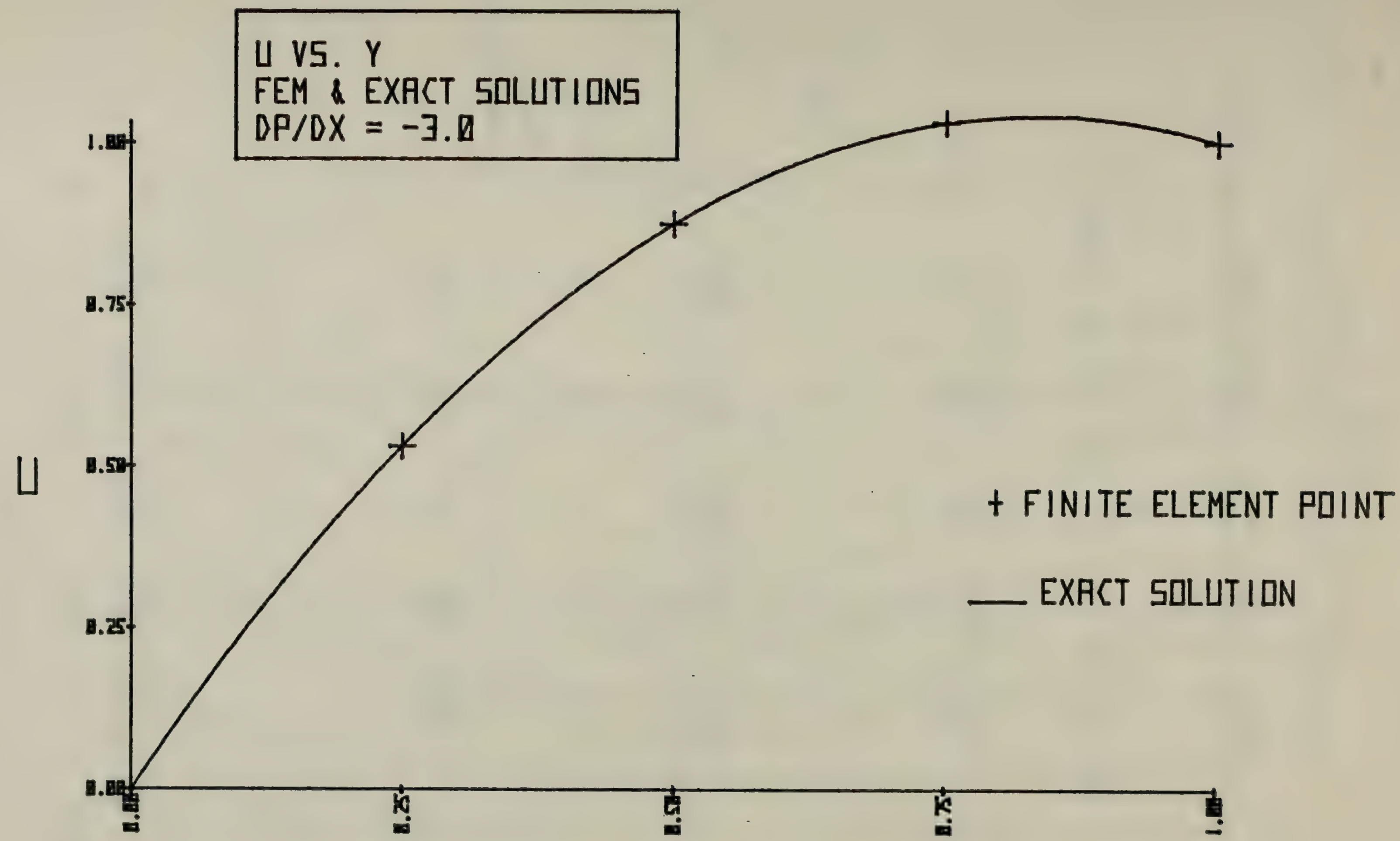
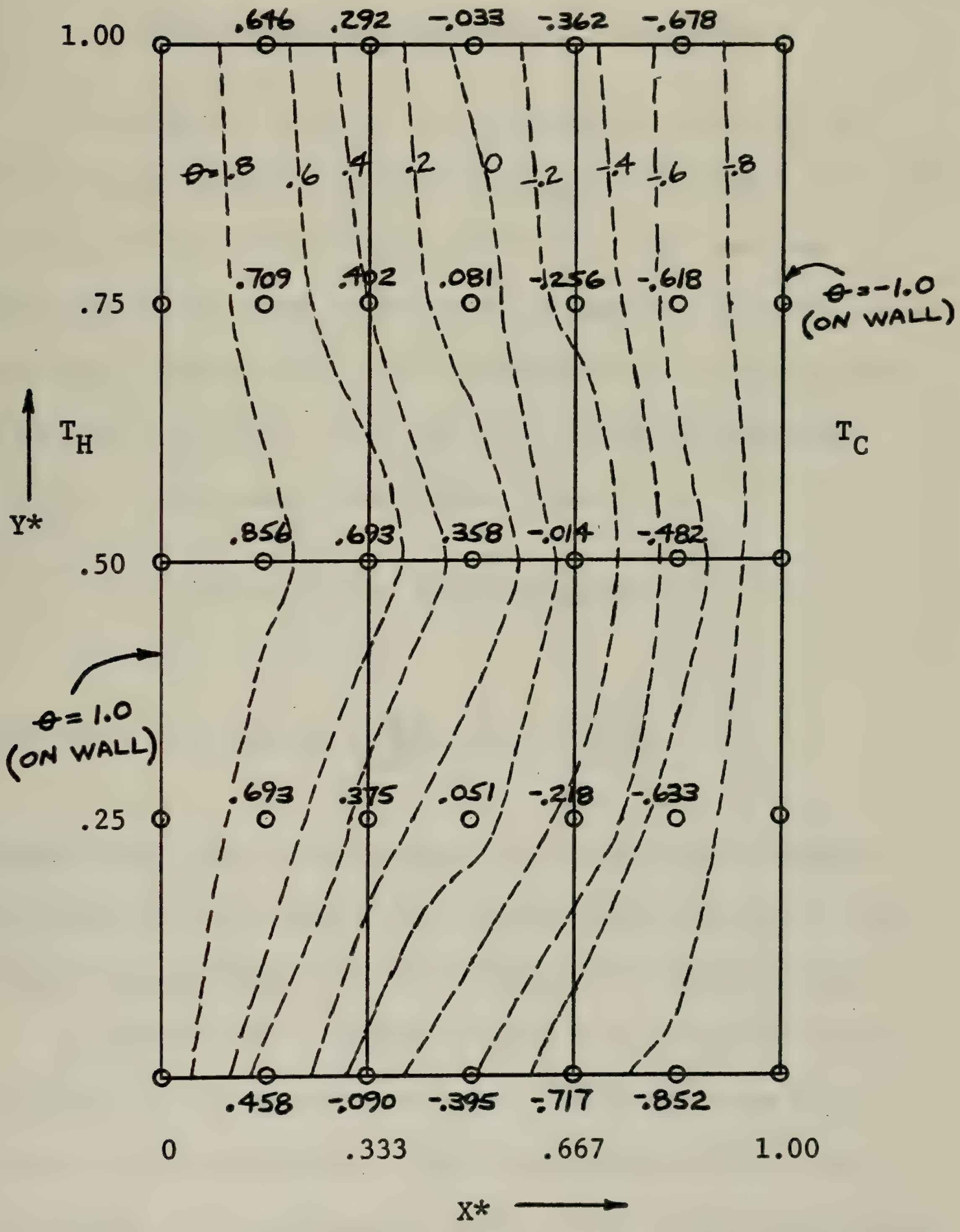


FIGURE 15
VELOCITY PROFILE FOR LINEAR COUETTE FLOW

Figure 16
Steady State Isotherms
(Nonlinear Heat Transfer Problem)



Gr_L=9.464x10^2, Pr=1.0755x10^4, L/D=4.53

APPENDIX B

BRIEF REVIEW ON CALCULUS OF VARIATIONS

A fundamental problem in differential calculus is extremizing (maximizing or minimizing) a function f(x) for a range of the independent variable x. The problem in variational calculus is also extremization; however, it is concerned chiefly with the extremization of a functional. A simple functional, in terms of only one independent variable, would have the typical form

$$I(\phi) = \int_{x_1}^{x_2} F(x, \phi, \phi_x, \phi_{xx})dx$$

where $\phi = \phi(x)$ and $\phi_x = \frac{\partial \phi}{\partial x}$, $\phi_{xx} = \frac{\partial^2 \phi}{\partial x^2}$.

Summarizing, the two branches of calculus are related in that both are concerned with an extremum; one deals with number spaces while the other deals with function spaces.

In variational problems a functional which is characteristic of the problem is first formed in terms of a function (or functions). Then variations of this same functional are investigated with a view toward extremizing the functional. In some cases this approach results in a

closed form, exact solution. But more often, the problem must be solved by an approximate method. One such method is the Rayleigh-Ritz technique. This approach is preferable to the direct application of finite difference methodology to solve the differential equation with its associated boundary conditions, because the functional can often be used to assure convergence of the approximate solution.

A simple example of variational calculus is the problem of finding the plane curve joining two points (x_1, y_1) and (x_2, y_2) which has the shortest length. The solution sought here is the function y(x) describing the curve of shortest length; the corresponding functional is the length of the curve given by

$$I(y) = \int_{x_1}^{x_2} \sqrt{1+\left(\frac{dy}{dx}\right)^2} \, dx$$

Using the method of variation of calculus implies that of all the curves

$$Y(x) = y(x) + \varepsilon\gamma(x)$$

which pass through the given end points, the shortest one y(x) must be selected. The problem thus reduces to finding the function y(x) that makes the integral I(y) a minimum.

Generally, in order to minimize the integral

$$I(y) = \int_{x_1}^{x_2} F(x, y, y')dx$$

where $y' = \frac{dy}{dx}$, the function $y(x)$ must satisfy the boundary conditions and the Euler-Lagrange differential equation

$$\frac{\partial F}{\partial y} - \frac{d}{dx}\left(\frac{\partial F}{\partial y'}\right) = 0$$

The previous result could be extended to several dependent and independent variables. For example, in order to minimize the integral

$$I(\phi) = \iint_A F(x, y, \phi, \phi_x, \phi_y)dxdy$$

in which ϕ_x and ϕ_y are the partial derivatives of ϕ with respect to x and y, respectively, the general function ϕ must satisfy the Euler-Lagrange differential equation

$$\frac{\partial F}{\partial \phi} - \frac{\partial}{\partial x}\left(\frac{\partial F}{\partial \phi_x}\right) - \frac{\partial}{\partial y}\left(\frac{\partial F}{\partial \phi_y}\right) = 0$$

in addition to the specified boundary conditions.

In the past two decades, since the advent of high speed digital computers, the variational formulation has been quite extensively employed in the fields of structural

and continuum mechanics. Important variational principles such as least work, minimum strain energy, minimum potential energy, and Reissner's variational theorem of elasticity have been well developed and are documented in standard textbooks. However, similar variational principles applicable to fluid mechanic problems have not been as comprehensively developed. Calculus of variations has, until only recently, been utilized sparingly in the field of fluid mechanics.

THIS IS A 2-D NONLINEAR COUETTE FLOW PROBLEM

IBAND= 26

NEQ = 82

NO. OF NODES= 35

NO. OF ELEMENTS= 12

NO. OF CORNER NODES= 12

NNVELS= 14

NNCXY= 21

NNPS= 12

SUMMARY OF NODAL COORDINATES

I	X(I)	Y(I)
1	0.0	1.000
3	0.0	0.500
5	0.0	0.0
11	0.333	1.000
13	0.333	0.500
15	0.333	0.0
21	0.667	1.000
23	0.667	0.500
25	0.667	0.0
31	1.000	1.000
33	1.000	0.500
35	1.000	0.0

LISTING OF SYSTEM TOPOLOGY

ELEMENT NUMBER	NODE NUMBERS					
1	1	7	13	12	11	6
2	1	2	3	8	13	7
3	3	4	5	9	13	8
4	5	10	15	14	13	9
5	11	17	23	22	21	16
6	11	12	13	18	23	17
7	13	14	15	19	23	18
8	15	20	25	24	23	19
9	21	27	33	32	31	26
10	21	22	23	28	33	27
11	23	24	25	29	33	28
12	25	30	35	34	33	29

NODES WHERE VELOCITIES ARE SPECIFIED

I	NODE	U VELOCITY	V VELOCITY
1	1	1.000	0.0
2	5	0.0	0.0
3	6	1.000	0.0
4	10	0.0	0.0
5	11	1.000	0.0
6	15	0.0	0.0
7	16	1.000	0.0
8	20	0.0	0.0
9	21	1.000	0.0
10	25	0.0	0.0
11	26	1.000	0.0
12	30	0.0	0.0
13	31	1.000	0.0
14	35	0.0	0.0

NODES WHERE QX AND QY ARE SPECIFIED

I	NODE	QX	QY
1	7	0.0	0.0
2	8	0.0	0.0
3	9	0.0	0.0
4	12	0.0	0.0
5	13	0.0	0.0
6	14	0.0	0.0
7	17	0.0	0.0
8	18	0.0	0.0
9	19	0.0	0.0
10	22	0.0	0.0
11	23	0.0	0.0
12	24	0.0	0.0
13	27	0.0	0.0
14	28	0.0	0.0
15	29	0.0	0.0
16	2	1.333	0.0
17	3	0.667	0.0
18	4	1.333	0.0
19	32	-0.333	0.0
20	33	-0.167	0.0
21	34	-0.333	0.0

NODES WHERE PRESSURE IS SPECIFIED

I	NODE	PRESSURE
1	1	4.000
2	3	4.000
3	5	4.000
4	11	3.000
5	13	3.000
6	15	3.000
7	21	2.000
8	23	2.000
9	25	2.000
10	31	1.000
11	33	1.000
12	35	1.000

NODAL VARIABLE IS THE U-VELOCITY AT NODES 1 - 35;
THE V-VELOCITY AT NODES 36 - 70;
AND THE PRESSURES AT NODES 71 - 82.

THE FIRST SEQUENCE OF 82 NODAL VARIABLES
REPRESENTS A LINEAR, STEADY STATE SYSTEM;
WHILE THE SECOND SET OF THE 82 VALUES CORRESPONDS
TO A NONLINEAR ANALYSIS OF THE SOLUTION DOMAIN.

A PRESSURE GRADIENT IN THE HORIZONTAL SHEAR
DIRECTION OF -3 UNITS IN MAGNITUDE HAS BEEN ADDED
TO PRODUCE A CURVED VELOCITY PROFILE.

NODE NO.	NODE VARIABLES
1	0.1000000000D 01
2	0.1031249996D 01
3	0.8750000067D 00
4	0.5312499961D 00
5	0.0
6	0.1000000000D 01
7	0.1031249999D 01
8	0.8750000022D 00
9	0.5312499991D 00
10	0.0
11	0.1000000000D 01
12	0.1031249999D 01
13	0.8749999958D 00
14	0.5312499955D 00
15	0.0
16	0.1000000000D 01
17	0.1031250000D 01
18	0.8749999958D 00
19	0.5312499999D 00
20	0.0
21	0.1000000000D 01
22	0.1031250002D 01
23	0.8750000014D 00
24	0.5312500020D 00
25	0.0
26	0.1000000000D 01
27	0.1031250006D 01
28	0.8749999986D 00
29	0.5312500056D 00
30	0.0
31	0.1000000000D 01
32	0.1031250014D 01
33	0.8749999887D 00
34	0.5312500139D 00
35	0.0
36	0.0
37	0.6099426964D-19
38	-0.7125187818D-42
39	-0.6099426964D-19
40	0.0
41	0.0
42	0.8810283338D-19

43	-0.8133546689D-42
44	-0.8810283338D-19
45	0.0
46	0.0
47	0.1935250252D-18
48	-0.1114902041D-41
49	-0.1935250252D-18
50	0.0
51	0.0
52	0.4709694692D-18
53	-0.1571832282D-41
54	-0.4709694692D-18
55	0.0
56	0.0
57	0.1167053458D-17
58	-0.2186840317D-41
59	-0.1167053458D-17
60	0.0
61	0.0
62	0.2900518273D-17
63	-0.2671579697D-41
64	-0.2900518273D-17
65	0.0
66	0.0
67	0.2008051124D-17
68	-0.3356588546D-41
69	-0.2008051124D-17
70	0.0
71	0.4000000000D 01
72	0.4000000000D 01
73	0.4000000000D 01
74	0.3000000000D 01
75	0.3000000000D 01
76	0.3000000000D 01
77	0.2000000000D 01
78	0.2000000000D 01
79	0.2000000000D 01
80	0.1000000000D 01
81	0.1000000000D 01
82	0.1000000000D 01

NODE NO.	NODE VARIABLES
1	0.1000000000D 01

2	0.1028887453D 01
3	0.8689584459D 00
4	0.5288644394D 00
5	0.0
6	0.1000000000D 01
7	0.1029118812D 01
8	0.8708250523D 00
9	0.5291035809D 00
10	0.0
11	0.1000000000D 01
12	0.1029300447D 01
13	0.8707893553D 00
14	0.5292824849D 00
15	0.0
16	0.1000000000D 01
17	0.1029562570D 01
18	0.8718195572D 00
19	0.5295482546D 00
20	0.0
21	0.1000000000D 01
22	0.1029726484D 01
23	0.8714843669D 00
24	0.5297096677D 00
25	0.0
26	0.1000000000D 01
27	0.1030036890D 01
28	0.8728957661D 00
29	0.5300233390D 00
30	0.0
31	0.1000000000D 01
32	0.1030173492D 01
33	0.8734199036D 00
34	0.5301609830D 00
35	0.0
36	0.0
37	0.9409332543D-19
38	0.2406344091D-19
39	-0.5690421895D-19
40	0.0
41	0.0
42	0.1282493313D-18
43	0.2611072288D-19
44	-0.8653878326D-19
45	0.0

46	0.0
47	0.2538354935D-18
48	0.3200619005D-19
49	-0.1998692334D-18
50	0.0
51	0.0
52	0.5613469984D-18
53	0.3635113503D-19
54	-0.4911924371D-18
55	0.0
56	0.0
57	0.1281917568D-17
58	0.3780625767D-19
59	-0.1200746469D-17
60	0.0
61	0.0
62	0.2962239092D-17
63	0.3211763376D-19
64	-0.2910845719D-17
65	0.0
66	0.0
67	0.2008289764D-17
68	0.1510271317D-19
69	-0.1990889434D-17
70	0.0
71	0.4000000000D 01
72	0.4000000000D 01
73	0.4000000000D 01
74	0.3000000000D 01
75	0.3000000000D 01
76	0.3000000000D 01
77	0.2000000000D 01
78	0.2000000000D 01
79	0.2000000000D 01
80	0.1000000000D 01
81	0.1000000000D 01
82	0.1000000000D 01

STEADY STATE FLUID MECHANICS PROBLEM

(HEAT TRANSFER)

THIS IS A 2-D NONLINEAR PROBLEM

IBAND= 26

NEC=117

NO. OF NODES= 35

NO. OF ELEMENTS= 12

NO. OF CORNER NODES= 12

NNVELS= 20

NNCXY= 15

NNPS= 6

NNTS= 10

NNCZC= 6

NNCZ= 25

SUMMARY OF NODAL COORDINATES

I	X(I)	Y(I)
1	0.0	8.500
3	0.0	4.250
5	0.0	0.0
11	0.625	8.500
13	0.625	4.250
15	0.625	0.0
21	1.250	8.500
23	1.250	4.250
25	1.250	0.0
31	1.875	8.500
33	1.875	4.250
35	1.875	0.0

LISTING OF SYSTEM TOPOLOGY

ELEMENT NUMBER	NODE NUMBERS					
1	1	7	13	12	11	6
2	1	2	3	8	13	7
3	3	4	5	9	13	8
4	5	10	15	14	13	9
5	11	17	23	22	21	16
6	11	12	13	18	23	17
7	13	14	15	19	23	18
8	15	20	25	24	23	19
9	21	27	33	32	31	26
10	21	22	23	28	33	27
11	23	24	25	29	33	28
12	25	30	35	34	33	29

NODES WHERE VELOCITIES ARE SPECIFIED

I NODE U VELOCITY V VELOCITY

1	1	0.0	0.0
2	2	0.0	0.0
3	3	0.0	0.0
4	4	0.0	0.0
5	5	0.0	0.0
6	6	0.0	0.0
7	10	0.0	0.0
8	11	0.0	0.0
9	15	0.0	0.0
10	16	0.0	0.0
11	20	0.0	0.0
12	21	0.0	0.0
13	25	0.0	0.0
14	26	0.0	0.0
15	30	0.0	0.0
16	31	0.0	0.0
17	32	0.0	0.0
18	33	0.0	0.0
19	34	0.0	0.0
20	35	0.0	0.0

NODES WHERE QX AND QY ARE SPECIFIED

I	NODE	QX	QY
1	7	0.0	0.0
2	8	0.0	0.0
3	9	0.0	0.0
4	12	0.0	0.0
5	13	0.0	0.0
6	14	0.0	0.0
7	17	0.0	0.0
8	18	0.0	0.0
9	19	0.0	0.0
10	22	0.0	0.0
11	23	0.0	0.0
12	24	0.0	0.0
13	27	0.0	0.0
14	28	0.0	0.0
15	29	0.0	0.0

NODES WHERE PRESSURE IS SPECIFIED

I	NODE	PRESSURE
1	1	1014000.000
2	3	1014000.000
3	5	1014000.000
4	31	1014000.000
5	33	1014000.000
6	35	1014000.000

NODES WHERE TEMPERATURE IS SPECIFIED

I	NODE	TEMPERATURE
1	1	25.000
2	2	25.000
3	3	25.000
4	4	25.000
5	5	25.000
6	31	20.000
7	32	20.000
8	33	20.000
9	34	20.000
10	35	20.000

NODES WHERE QZC IS SPECIFIED

I	NODE	QZC
1	4	0.0
2	5	0.0
3	6	0.0
4	7	0.0
5	8	0.0
6	9	0.0

NODES WHERE HEAT FLUX QZ IS SPECIFIED

I	NODE	HEAT FLUX
1	6	0.0
2	7	0.0
3	8	0.0
4	9	0.0
5	10	0.0
6	11	0.0
7	12	0.0
8	13	0.0
9	14	0.0
10	15	0.0
11	16	0.0
12	17	0.0
13	18	0.0
14	19	0.0
15	20	0.0
16	21	0.0
17	22	0.0
18	23	0.0
19	24	0.0
20	25	0.0
21	26	0.0
22	27	0.0
23	28	0.0
24	29	0.0
25	30	0.0

NODAL VARIABLE IS THE U-VELOCITY AT NODES 1 - 35;
THE V-VELOCITY AT NODES 36 - 70;
THE PRESSURE AT NODES 71 - 82;
AND THE TEMPERATURE AT NODES 83 - 117.

THE SPECIFIED WALL PRESSURES
ARE NORMALIZED TO ONE (1) ATMOSPHERE,
THAT IS, 1014000 DYNES/SQ.CM
(ALL PARAMETER VALUES ARE IN CGS UNITS).

THE FIRST SEQUENCE OF 117 NODAL VARIABLES
REPRESENTS A LINEAR,STEADY STATE SYSTEM;
WHILE THE SECOND SET OF THE 117 VALUES CORRESPONDS
TO A NONLINEAR ANALYSIS OF THE SOLUTION DOMAIN.

NODE NO.	NODE VARIABLE
1	0.0
2	0.0
3	0.0
4	0.0

5	0.0
6	0.0
7	-0.1594580172D 01
8	-0.7882672787D 01
9	-0.1594580172D 01
10	0.0
11	0.0
12	-0.3649907488D 00
13	-0.1203896378D 02
14	-0.3649907488D 00
15	0.0
16	0.0
17	-0.1216689153D 00
18	-0.9223308785D 01
19	-0.1216689153D 00
20	0.0
21	0.0
22	-0.1605186515D 01
23	-0.1070997583D 02
24	-0.1605186515D 01
25	0.0
26	0.0
27	-0.1686613562D 01
28	-0.6458410240D 01
29	-0.1686613562D 01
30	0.0
31	0.0
32	0.0
33	0.0
34	0.0
35	0.0
36	0.0
37	0.0
38	0.0
39	0.0
40	0.0
41	0.0
42	0.3287813820D 00
43	-0.8671505203D-23
44	-0.3287813820D 00
45	0.0
46	0.0
47	0.4841816743D 00
48	-0.1212888826D-22

49	-0.4041818743D 00
50	0.0
51	0.0
52	0.2853206050D 00
53	0.4210919668D-23
54	-0.2853206050D 00
55	0.0
56	0.0
57	0.1589243707D-01
58	0.9241557484D-23
59	-0.1589243707D-01
60	0.0
61	0.0
62	-0.4346077697D-01
63	0.1120835943D-22
64	0.4346077697D-01
65	0.0
66	0.0
67	0.0
68	0.0
69	0.0
70	0.0
71	0.1014000000D 07
72	0.1014000000D 07
73	0.1014000000D 07
74	0.1014037064D 07
75	0.1014225657D 07
76	0.1014037064D 07
77	0.1013980313D 07
78	0.1013875067D 07
79	0.1013980313D 07
80	0.1014000000D 07
81	0.1014000000D 07
82	0.1014000000D 07
83	0.2500000000D 02
84	0.2500000000D 02
85	0.2500000000D 02
86	0.2500000000D 02
87	0.2500000000D 02
88	0.2416666667D 02
89	0.2416666667D 02
90	0.2416666667D 02
91	0.2416666667D 02
92	0.2416666667D 02

93	0.2333333333D 02
94	0.2333333333D 02
95	0.2333333333D 02
96	0.2333333333D 02
97	0.2333333333D 02
98	0.2250000000D 02
99	0.2250000000D 02
100	0.2250000000D 02
101	0.2250000000D 02
102	0.2250000000D 02
103	0.2166666667D 02
104	0.2166666667D 02
105	0.2166666667D 02
106	0.2166666667D 02
107	0.2166666667D 02
108	0.2083333333D 02
109	0.2083333333D 02
110	0.2083333333D 02
111	0.2083333333D 02
112	0.2083333333D 02
113	0.2000000000D 02
114	0.2000000000D 02
115	0.2000000000D 02
116	0.2000000000D 02
117	0.2000000000D 02

NODE NO.	NODE VARIABLE
1	0.0
2	0.0
3	0.0
4	0.0
5	0.0
6	0.0
7	-0.1671967327D 01
8	-0.8146388287D 01
9	-0.1667709024D 01
10	0.0
11	0.0
12	-0.3529066201D 00
13	-0.1267086943D 02
14	-0.3464796968D 00
15	0.0
16	0.0

17	-0.9554759755D-01
18	-0.9662386104D 01
19	-0.8874654653D-01
20	0.0
21	0.0
22	-0.1643794550D 01
23	-0.1118836761D 02
24	-0.1639374227D 01
25	0.0
26	0.0
27	-0.1666366023D 01
28	-0.6868055602D 01
29	-0.1659751782D 01
30	0.0
31	0.0
32	0.0
33	0.0
34	0.0
35	0.0
36	0.0
37	0.0
38	0.0
39	0.0
40	0.0
41	0.0
42	0.3928000344D 00
43	0.2871715127D-03
44	-0.3916760591D 00
45	0.0
46	0.0
47	0.5930360944D 00
48	0.6126243490D-03
49	-0.5907199384D 00
50	0.0
51	0.0
52	0.3767079743D 00
53	0.9960375406D-03
54	-0.3731421378D 00
55	0.0
56	0.0
57	0.6382715612D-01
58	0.1376049903D-02
59	-0.5987522856D-01
60	0.0

61	0.0
62	-0.2429545074D-01
63	0.1210812751D-02
64	0.2677263620D-01
65	0.0
66	0.0
67	0.0
68	0.0
69	0.0
70	0.0
71	0.1014000000D 07
72	0.1014000000D 07
73	0.1014000000D 07
74	0.1014038420D 07
75	0.1014261630D 07
76	0.1014038452D 07
77	0.1013995776D 07
78	0.1013884020D 07
79	0.1014000754D 07
80	0.1014000000D 07
81	0.1014000000D 07
82	0.1014000000D 07
83	0.2500000000D 02
84	0.2500000000D 02
85	0.2500000000D 02
86	0.2500000000D 02
87	0.2500000000D 02
88	0.2411490471D 02
89	0.2427187487D 02
90	0.2463978352D 02
91	0.2423258019D 02
92	0.2364452284D 02
93	0.2323056484D 02
94	0.2350586481D 02
95	0.2423194982D 02
96	0.2343829368D 02
97	0.2227636650D 02
98	0.2241562105D 02
99	0.2270150631D 02
100	0.2339548232D 02
101	0.2262750919D 02
102	0.2151348362D 02
103	0.2159431099D 02
104	0.2185937901D 02

105	0.2246590917D 02
106	0.2179546544D 02
107	0.2070833022D 02
108	0.2080608577D 02
109	0.2095508338D 02
110	0.2129443533D 02
111	0.2091829352D 02
112	0.2037035832D 02
113	0.2000000000D 02
114	0.2000000000D 02
115	0.2000000000D 02
116	0.2000000000D 02
117	0.2000000000D 02

THE KINEMATIC VISCOSITY OF FLUID 50-HB-3520 AT 20 DEGREES C. =
10.90 SC.CM/SEC

THE DENSITY OF 50-HB-3520 AT 20 DEGREES C. = 1.0596 GM/CC

THE COEFF. OF THERMAL EXPANSION OF 50-HB-3520 AT 20 DEGREES C. =
0.002278/DEGREE C.

THE THERMAL DIFFUSIVITY OF 50-HB-3520 AT 20 DEGREES C. =
0.00103 SQ.CM/SEC

THE GRASHOF NUMBER (GR(L)) =(G*B*L**3*(TH-TC))/V**2 = 946.4

THE U VELOCITY FORCING FUNCTION,G*B*T(INITIAL), = 65.962 CM/SQ.S

STEADY STATE COUETTE FLOW PRCBLEM

```
      IMPLICIT REAL*8(A-H,O-Z,$)                                        COUT0001
C   THE U VELOCITY IS IN THE FIRST NN POSITIONS OF T(I)  (NN=NO. OF NODESCOUT0002
C   THE V VELOCITY IS IN THE SECOND NN POSITIONS OF T I.E. T(I+NN)      COUT0003
C   THE P PRESSURE IS IN THE NN+NN+I POSITIONS OF T  I.E. T(I+NN+NN)    COUT0004
C   THERE ARE NNCN PRESSURE NODES                                       COUT0005
C                                                                       COUT0006
C                                                                       COUT0007
      DATA NREAD/5/                                                     COUT0008
      DATA NWRITE/6/                                                    COUT0009
      DATA STOP/'STOP'/                                                 COUT0010
                                                                        COUT0011
      DIMENSION TM(82,82)                                               COUT0012
      DIMENSION XC(82), YC(82), NODE(82,6), NVS(82), NCN(82)            COUT0013
      DIMENSION T(82), NVIS(82), NCP(82), NPS(82), Q(82)                COUT0014
      DIMENSION NQS(82), T1(82)                                         COUT0015
      DIMENSION TM$(15,15), N(15), NQIS(82)                             COUT0016
      DIMENSION RHS(82)                                                 COUT0017
      DIMENSICN RP$(6), ZP$(6)                                          COUT0018
      DIMENSION XC$(3), YC$(3), WKAREA(12000)                           COUT0019
C                                                                       COUT0020
C   THE FIRST PART OF THE PROGRAM CAN BE CONSIDERED AS AN INPUT ROUTINE COUT0021
C   LINE 014 TO  0151 ARE INPUT AND VERIFICATION OF ALL DATA            COUT0022
C                                                                       COUT0023
C   THIS WOULD BE PART OF ANY FINITE ELEMENT PROGRAM                    COUT0024
C                                                                       COUT0025
C                                                                       COUT0026
C   READ IN NUMBER OF NODES AND ELEMENTS AND NO. OF CORNER NODES        COUT0027
      READ (NREAD,3800) NN,NE,NNCN                                      COUT0028
      WRITE(NWRITE,1094)                                                COUT0029
C                                                                       COUT0030
C     INITIALIZE ALL PARAMETERS                                         COUT0031
C                                                                       COUT0032
      MM = 2*NN+NNCN                                                    COUT0033
C                                                                       COUT0034
      DO 200 I=1,MM                                                     COUT0035
      XC(I) = 0.D0                                                      COUT0036
```

```
      YC(I) = 0.D0
      NVS(I) = 0
      NCN(I) = 0
      NCP(I) = 0
      NPS(I) = 0
      NQS(I) = 0
C
      DO 200 J=1,6
      NODE(I,J) = 0
  200 CONTINUE
C
C
      DO 300 I=1,MM
      T1(I) = 0.D0
      T(I) = 0.D0
      NVIS(I) = 0
      Q(I) = 0.D0
      NQIS(I) = 0
      RHS(I) = 0.D0
C
      DO 300 J=1,MM
      TM(I,J) = 0.D0
  300 CONTINUE
C
C
      DO 400 I=1,15
      N(I) = 0
C
      DO 400 J=1,15
      TM$(I,J) = 0.D0
  400 CONTINUE
C
C
      DO 500 I=1,6
      RP$(I) = 0.D0
      ZP$(I) = 0.D0
  500 CONTINUE
C
C
      DO 600 I=1,3
      XC$(I) = 0.D0
      YC$(I) = 0.D0
  600 CONTINUE
C
C    REAC NODE NUMBERS AND COORDINATES
C
C
      DO 700 J=1,NN
```

```
      READ (NREAD,3900) WORD,I,XC(I),YC(I)
      IF (WORD.EQ.STOP) GO TO 800
      NCN(J) = I
  700 CONTINUE
C
  800 NNCN = J-1
C
C   THE ARRAY  NCP(J) GENERATES THE GLOBAL PRESSURE INDICES (P1,P2.)
C   THUS PRESSURE NODES ARE LABELED AS CORNER NODES ARE INPUTTED
C   WHEN ONE INPUTS A GLOBAL CORNER NODE FOR J
C
C
      DO 900 J=1,NNCN
      NCP(NCN(J)) = J+NN+NN
  900 CONTINUE
C
C
C    SYSTEM TOPOLOGY( ELEMENT NO. AND NODE NUMBERS IN
C  COUNTER-CLOCKWISE FASHION STARTING AT ANY CORNER NODE
C  ALWAYS COUNT FROM UPPER LEFT HAND CORNER
C
C
      DO 1000 I=1,NE
      READ (NREAD,4000) J,NODE(J,1),NODE(J,2),NODE(J,3),NODE(J,4),NODE(J
     1,5),NODE(J,6)
 1000 CONTINUE
C
      MAXDIF = 0
C
      DO 1100 I=1,NE
C
      DO 1100 J=1,6
C
      DO 1100 K=1,6
      LL = IABS(NODE(I,J)-NODE(I,K))
      IF (LL.GT.MAXDIF) MAXDIF=LL
      IBAND = 2*(MAXDIF+1)
      NEQ = 2*NN+NNCN
 1100 CONTINUE
C
      WRITE (NWRITE,3700) IBAND,NEQ
C
C   READ NODES WHERE BOTH U AND V VELOCITY IS SPECIFIED
C
C
      DO 1200 I=1,MM
      READ (NREAD,3900) WORD,NVELS,VELU,VELV
      IF (WORD.EQ.STOP) GO TO 1300
```

```
      NVS(I) = NVELS
      T(NVS(I)) = VELU
      T(NVS(I)+NN) = VELV
 1200 CONTINUE
C
C
C  COUNT NODES HAVING SPECIFIED  VELOCITIES
C
 1300 NNVELS = I-1
C
C
C READ QX AND QY VALUES AT INTERNAL NODES
C
      DO 1400 I=1,NN
      READ (NREAD,3900) WORD,NQXY,QXNS,QYNS
      IF (WORD.EQ.STOP) GO TO 1500
      NQS(I) = NQXY
      Q(NQS(I)) = QXNS
      Q(NQS(I)+NN) = QYNS
 1400 CCNTINUE
C
C
C  COUNT NODES HAVING SPECIFIED QX AND QY
C
 1500 NNQXY = I-1
C
C  READ NODE NUMBER AND PRESSURE WHERE SPECIFIED
C
C
      DO 1600 I=1,NN
      READ (NREAD,4100) WORD,NP,PNP
      IF (WORD.EQ.STOP) GO TO 1700
      NPS(I) = NP
      T(NCP(NPS(I))) = PNP
 1600 CONTINUE
C
C
C COUNT  BOUNDARY NODES WHERE PRESSURE SPECIFIED
C
 1700 NNPS = I-1
C
C NQIS IS A LIST OF THE INDICES OF KNOWN QX,QY
C
      DC 2000 I=1,NNQXY
      NQIS(I) = NQS(I)
      NQIS(I+NNQXY) = NQS(I)+NN
 2000 CCNTINUE
C
```

```
C
C
C  NVIS IS A LIST OF KNOWN VELOCITY AND PRESSURE INDICES
C
      DO 2200 I=1,NNVELS
      NVIS(I) = NVS(I)
      NVIS(I+NNVELS) = NVS(I)+NN
 2200 CONTINUE
C
C
      DO 2300 J=1,NNPS
      NVIS(2*NNVELS+J) = NCP(NPS(J))
 2300 CONTINUE
C
C NNHC=NUMBER OF NODES WHERE HEAT TRANSFER COEFFICIENT IS SPECIFIED
      NNHC = 0
C
C  NTOTQ=TOTAL NUMBER OF KNOWN QX,QY
C
      NTOTQ = 2*NNQXY
C
C NTOTVP=TOTAL NUMBER OF KNOWN VELOCITIES, AND PRESSURES
C
      NTOTVP = 2*NNVELS+NNPS
C
C  PRINT ALL INPUT DATA
C
      WRITE (NWRITE,4300) NN,NE,NNCN
      WRITE (NWRITE,4400) NNVELS
      WRITE (NWRITE,4500) NNQXY
      WRITE (NWRITE,4600) NNPS
      WRITE (NWRITE,4700)
C
      DO 2400 I=1,NNCN
      WRITE (NWRITE,4800) NCN(I),XC(NCN(I)),YC(NCN(I))
 2400 CONTINUE
C
      WRITE (NWRITE,4900)
C
      DO 2500 I=1,NE
      WRITE (NWRITE,5000) I,NODE(I,1),NODE(I,2),NODE(I,3),NODE(I,4),NODE
     1(I,5),NODE(I,6)
 2500 CONTINUE
C
      WRITE (NWRITE,5100)
C
      DO 2600 I=1,NNVELS
      WRITE (NWRITE,5200) I,NVS(I),T(NVS(I)),T(NVS(I)+NN)
```

```
 2600 CONTINUE
C
      WRITE (NWRITE,5300)
C
      DO 2700 I=1,NNQXY
      WRITE (NWRITE,5200) I,NQS(I),Q(NQS(I)),Q(NQS(I)+NN)
 2700 CONTINUE
C
      WRITE (NWRITE,5400)
C
      DO 2800 I=1,NNPS
      WRITE (NWRITE,5600) I,NPS(I),T(NCP(NPS(I)))
 2800 CONTINUE
      WRITE(NWRITE,1090)
      WRITE(NWRITE,1092)
      WRITE(NWRITE,1093)
 2840 CONTINUE
      DO 2850 I=1,MM
      RHS(I) = 0.D0
      DO 2850 J=1,MM
      TM(I,J) = 0.D0
 2850 CONTINUE
      DO 2860 I=1,15
      DO 2860 J=1,15
      TM$(I,J) = 0.D0
 2860 CONTINUE
C
C
C    END OF  INPUT AND VERIFICATION ROUTINE
C
C
      DO 3200 K=1,NE
      N1=NODE(K,1)
      N2=NODE(K,2)
      N3=NODE(K,3)
      N4=NODE(K,4)
      N5=NODE(K,5)
      N6=NODE(K,6)
      N7=NODE(K,1)+NN
      N8=NODE(K,2)+NN
      N9=NODE(K,3)+NN
      N10=NODE(K,4)+NN
      N11=NODE(K,5)+NN
      N12=NODE(K,6)+NN
      N13=NCP(NODE(K,1))
      N14=NCP(NODE(K,3))
      N15=NCP(NODE(K,5))
```

```
      XC$(1) = XC(NODE(K,1))                                            COUT0277
      XC$(2) = XC(NODE(K,3))                                            COUT0278
      XC$(3) = XC(NODE(K,5))                                            COUT0279
      YC$(1) = YC(NODE(K,1))                                            COUT0280
      YC$(2) = YC(NODE(K,3))                                            COUT0281
      YC$(3) = YC(NODE(K,5))                                            COUT0282
      A = 1.D0                                                          COUT0283
      AA = 1.D0                                                         COUT0284
      A1 = XC$(2)*YC$(3)-XC$(3)*YC$(2)                                  COUT0285
      A2 = XC$(3)*YC$(1)-XC$(1)*YC$(3)                                  COUT0286
      A3 = XC$(1)*YC$(2)-XC$(2)*YC$(1)                                  COUT0287
      B1 = YC$(2)-YC$(3)                                                COUT0288
      B2 = YC$(3)-YC$(1)                                                COUT0289
      B3 = YC$(1)-YC$(2)                                                COUT0290
      C1 = XC$(3)-XC$(2)                                                COUT0291
      C2 = XC$(1)-XC$(3)                                                COUT0292
      C3 = XC$(2)-XC$(1)                                                COUT0293
      DEL = DABS(0.5D0*(XC$(1)*(YC$(2)-YC$(3))+XC$(2)*(YC$(3)-YC$(1))+XCCOUT0294
     1$(3)*(YC$(1)-YC$(2))))                                            COUT0295
      CONST = (1.D0*A/(3.D0*DEL))*AA                                    COUT0296
      D1=-B1/6.D0                                                       COUT0297
      D2=-B2/6.D0                                                       COUT0298
      D3=-B3/6.D0                                                       COUT0299
      E1=-C1/6.D0                                                       COUT0300
      E2=-C2/6.D0                                                       COUT0301
      E3=-C3/6.D0                                                       COUT0302
      F1=B1/2520.D0                                                     COUT0303
      F2=B2/2520.D0                                                     COUT0304
      F3=B3/2520.D0                                                     COUT0305
      G1=C1/2520.D0                                                     COUT0306
      G2=C2/2520.D0                                                     COUT0307
      G3=C3/2520.D0                                                     COUT0308
      U1=T1(N1)                                                         COUT0309
      U2=T1(N2)                                                         COUT0310
      U3=T1(N3)                                                         COUT0311
      U4=T1(N4)                                                         COUT0312
      U5=T1(N5)                                                         COUT0313
      U6=T1(N6)                                                         COUT0314
      V1=T1(N7)                                                         COUT0315
      V2=T1(N8)                                                         COUT0316
      V3=T1(N9)                                                         COUT0317
      V4=T1(N10)                                                        COUT0318
      V5=T1(N11)                                                        COUT0319
      V6=T1(N12)                                                        COUT0320
      TM$(1,1) = 0.75D0*(B1*B1+C1*C1)*CONST                             COUT0321
      TM$(1,2) = (B1*B2+C1*C2)*CONST                                    COUT0322
      TM$(1,3) = -TM$(1,2)*0.25D0                                       COUT0323
      TM$(1,4) = 0.                                                     COUT0324
```

```
      TM$(1,6) = (B1*B3+C1*C3)*CONST                                    COUT0325
      TM$(1,5) = -TM$(1,6)*.25D0                                        COUT0326
      TM$(3,3) = .75D0*(B2*B2+C2*C2)*CONST                              COUT0327
      TM$(3,4) = (B2*B3+C2*C3)*CONST                                    COUT0328
      TM$(3,5) = -TM$(3,4)*0.25D0                                       COUT0329
      TM$(2,1) = TM$(1,2)                                               COUT0330
      TM$(2,3) = TM$(1,2)                                               COUT0331
      TM$(2,2) = 8.D0/3.D0*(TM$(1,1)+TM$(3,3))+2.D0*TM$(1,2)           COUT0332
      TM$(2,4) = 2.D0*TM$(1,6)+TM$(3,4)+TM$(1,2)+4.D0/3.D0*TM$(3,3)    COUT0333
      TM$(2,5) = 0.                                                     COUT0334
      TM$(2,6) = TM$(1,6)+2.D0*TM$(3,4)+TM$(1,2)+4.D0/3.D0*TM$(1,1)    COUT0335
      TM$(3,1) = TM$(1,3)                                               COUT0336
      TM$(3,2) = TM$(2,3)                                               COUT0337
      TM$(3,6) = 0.                                                     COUT0338
      TM$(6,3) = TM$(3,6)                                               COUT0339
      TM$(5,5) = .75D0*(B3*B3+C3*C3)*CONST                              COUT0340
      TM$(4,1) = TM$(1,4)                                               COUT0341
      TM$(4,2) = TM$(2,4)                                               COUT0342
      TM$(4,3) = TM$(3,4)                                               COUT0343
      TM$(4,4) = 8.D0/3.D0*(TM$(3,3)+TM$(5,5))+2.D0*TM$(3,4)           COUT0344
      TM$(4,5) = TM$(3,4)                                               COUT0345
      TM$(4,6) = TM$(1,6)+TM$(3,4)+2.D0*TM$(1,2)+4.D0/3.D0*TM$(5,5)    COUT0346
      TM$(5,1) = TM$(1,5)                                               COUT0347
      TM$(5,2) = TM$(2,5)                                               COUT0348
      TM$(5,3) = TM$(3,5)                                               COUT0349
      TM$(5,4) = TM$(4,5)                                               COUT0350
      TM$(5,6) = TM$(1,6)                                               COUT0351
      TM$(6,1) = TM$(1,6)                                               COUT0352
      TM$(6,2) = TM$(2,6)                                               COUT0353
      TM$(6,4) = TM$(4,6)                                               COUT0354
      TM$(6,5) = TM$(5,6)                                               COUT0355
      TM$(6,6) = 8.D0/3.D0*(TM$(5,5)+TM$(1,1))+2.D0*TM$(1,6)           COUT0356
C                                                                       COUT0357
C     BEGIN INPUT OF NONLINEAR TERMS                                    COUT0358
C                                                                       COUT0359
      TM$(1,1)=TM$(1,1)                                                 COUT0360
     1-(78.D0*U1+48.D0*U2-9.D0*U3+12.D0*U4-9.D0*U5+48.D0*U6)*F1        COUT0361
     2              -(78.D0*V1+48.D0*V2-9.D0*V3+12.D0*V4-9.D0*V5+48.D0*V6)COUT0362
     3*G1                                                               COUT0363
      TM$(2,1)=TM$(2,1)                                                 COUT0364
     1-(48.D0*U1+160.D0*U2-32.D0*U3+16.D0*U4-20.D0*U5+80.D0*U6)*F1     COUT0365
     2              -(48.D0*V1+160.D0*V2-32.D0*V3+16.D0*V4-20.D0*V5+80.D0*COUT0366
     3V6)*G1                                                            COUT0367
      TM$(3,1)=TM$(3,1)                                                 COUT0368
     1-(-9.D0*U1-32.D0*U2-18.D0*U3-16.D0*U4+11.D0*U5-20.D0*U6)*F1      COUT0369
     2              -(-9.D0*V1-32.D0*V2--18.D0*V3--16.D0*V4+11.D0*V5-20.D0*VCOUT0370
     36)*G1                                                             COUT0371
      TM$(4,1)=TM$(4,1)                                                 COUT0372
```

```
1-(12.D0*U1+16.D0*U2-16.D0*U3-96.D0*U4-16.D0*U5+16.D0*U6)*F1        COUT0373
2             -(12.D0*V1+16.D0*V2-16.D0*V3-96.D0*V4-16.D0*V5+16.D0*VCOUT0374
36)*G1                                                              COUT0375
 TM$(5,1)=TM$(5,1)                                                  COUT0376
1-(-9.D0*U1-20.D0*U2+11.D0*U3-16.D0*U4-18.D0*U5-32.D0*U6)*F1        COUT0377
2             -(-9.D0*V1-20.D0*V2+11.D0*V3-16.D0*V4-18.D0*V5-32.D0*VCOUT0378
36)*G1                                                              COUT0379
 TM$(6,1)=TM$(6,1)                                                  COUT0380
1-(48.D0*U1+80.D0*U2-20.D0*U3+16.D0*U4-32.D0*U5+160.D0*U6)*F1       COUT0381
2             -(48.D0*V1+80.D0*V2-20.D0*V3+16.D0*V4-32.D0*V5+160.D0*COUT0382
3V6)*G1                                                             COUT0383
 TM$(1,2)=TM$(1,2)                                                  COUT0384
1-(24.D0*U1-32.D0*U2-16.D0*U3-48.D0*U4+4.D0*U5-16.D0*U6)*F1         COUT0385
2             -(24.D0*V1-32.D0*V2-16.D0*V3-48.D0*V4+4.D0*V5-16.D0*V6COUT0386
3)*G1                       -(120.D0*U1+48.D0*U2-16.D0*U3-16.D0*U4-16.COUT0387
4D0*U5+48.D0*U6)*F2                   -(120.D0*V1+48.D0*V2-16.D0*V3-COUT0388
516.D0*V4-16.D0*V5+48.D0*V6)*G2                                     COUT0389
 TM$(2,2)=TM$(2,2)                                                  COUT0390
1-(-32.D0*U1+384.D0*U2+48.D0*U3+192.D0*U4-48.D0*U5+128.D0*U6)*F1    COUT0391
2             -(-32.D0*V1+384.D0*V2+48.D0*V3+192.D0*V4-48.D0*V5+128.COUT0392
3D0*V6)*G1                  -(48.D0*U1+384.D0*U2-32.D0*U3+128.D0*U4-48COUT0393
4.D0*U5+192.D0*U6)*F2                 -(48.D0*V1+384.D0*V2-32.D0*V3+COUT0394
5128.D0*V4-48.D0*V5+192.D0*V6)*G2                                   COUT0395
 TM$(3,2)=TM$(3,2)                                                  COUT0396
1-(-16.D0*U1+48.D0*U2+120.D0*U3+48.D0*U4-16.D0*U5-16.D0*U6)*F1      COUT0397
2             -(-16.D0*V1+48.D0*V2+120.D0*V3+48.D0*V4-16.D0*V5-16.D0COUT0398
3*V6)*G1                    -(-16.D0*U1-32.D0*U2+24.D0*U3-16.D0*U4+4.DCOUT0399
40*U5-48.D0*U6)*F2                    -(-16.D0*V1-32.D0*V2+24.D0*V3-COUT0400
516.D0*V4+4.D0*V5-48.D0*V6)*G2                                      COUT0401
 TM$(4,2)=TM$(4,2)                                                  COUT0402
1-(-48.D0*U1+192.D0*U2+48.D0*U3+384.D0*U4-32.D0*U5+128.D0*U6)*F1    COUT0403
2             -(-48.D0*V1+192.D0*V2+48.D0*V3+384.D0*V4-32.D0*V5+128.COUT0404
3D0*V6)*G1                  -(-16.D0*U1+128.D0*U2-16.D0*U3+128.D0*U4-1COUT0405
46.D0*U5+128.D0*U6)*F2                -(-16.D0*V1+128.D0*V2-16.D0*V3COUT0406
5+128.D0*V4-16.D0*V5+128.D0*V6)*G2                                  COUT0407
 TM$(5,2)=TM$(5,2)                                                  COUT0408
1-(4.D0*U1-48.D0*U2-16.D0*U3-32.D0*U4+24.D0*U5-16.D0*U6)*F1         COUT0409
2             -(4.D0*V1-48.D0*V2-16.D0*V3-32.D0*V4+24.D0*V5-16.D0*V6COUT0410
3)*G1                       -(-16.D0*U1-48.D0*U2+4.D0*U3-16.D0*U4+24.DCOUT0411
40*U5-32.D0*U6)*F2                    -(-16.D0*V1-48.D0*V2+4.D0*V3-1COUT0412
56.D0*V4+24.D0*V5-32.D0*V6)*G2                                      COUT0413
 TM$(6,2)=TM$(6,2)                                                  COUT0414
1-(-16.D0*U1+128.D0*U2-16.D0*U3+128.D0*U4-16.D0*U5+128.D0*U6)*F1    COUT0415
2             -(-16.D0*V1+128.D0*V2-16.D0*V3+128.D0*V4-16.D0*V5+128.COUT0416
3D0*V6)*G1                  -(48.D0*U1+192.D0*U2-48.D0*U3+128.D0*U4-32COUT0417
4.D0*U5+384.D0*U6)*F2                 -(48.D0*V1+192.D0*V2-48.D0*V3+COUT0418
5128.D0*V4-32.D0*V5+384.D0*V6)*G2                                   COUT0419
 TM$(1,3)=TM$(1,3)                                                  COUT0420
```

```
1-(-18.D0*U1-32.D0*U2-9.D0*U3-20.D0*U4+11.D0*U5-16.D0*U6)*F2         COUT0421
2               -(-18.D0*V1-32.D0*V2-9.D0*V3-20.D0*V4+11.D0*V5-16.D0*VCOUT0422
36)*G2                                                                COUT0423
 TM$(2,3)=TM$(2,3)                                                    COUT0424
1-(-32.D0*U1+160.D0*U2+48.D0*U3+80.D0*U4-20.D0*U5+16.D0*U6)*F2        COUT0425
2               -(-32.D0*V1+160.D0*V2+48.D0*V3+80.D0*V4-20.D0*V5+16.D0COUT0426
3*V6)*G2                                                              COUT0427
 TM$(3,3)=TM$(3,3)                                                    COUT0428
1-(-9.D0*U1+48.D0*U2+78.D0*U3+48.D0*U4-9.D0*U5+12.D0*U6)*F2           COUT0429
2               -(-9.D0*V1+48.D0*V2+78.D0*V3+48.D0*V4-9.D0*V5+12.D0*V6COUT0430
3)*G2                                                                 COUT0431
 TM$(4,3)=TM$(4,3)                                                    COUT0432
1-(-20.D0*U1+80.D0*U2+48.D0*U3+160.D0*U4-32.D0*U5+16.D0*U6)*F2        COUT0433
2               -(-20.D0*V1+80.D0*V2+48.D0*V3+160.D0*V4-32.D0*V5+16.D0COUT0434
3*V6)*G2                                                              COUT0435
 TM$(5,3)=TM$(5,3)                                                    COUT0436
1-(11.D0*U1-20.D0*U2-9.D0*U3-32.D0*U4-18.D0*U5-16.D0*U6)*F2           COUT0437
2               -(11.D0*V1-20.D0*V2-9.D0*V3-32.D0*V4-18.D0*V5-16.D0*V6COUT0438
3)*G2                                                                 COUT0439
 TM$(6,3)=TM$(6,3)                                                    COUT0440
1-(-16.D0*U1+16.D0*U2+12.D0*U3+16.D0*U4-16.D0*U5-96.D0*U6)*F2         COUT0441
2               -(-16.D0*V1+16.D0*V2+12.D0*V3+16.D0*V4-16.D0*V5-96.D0*COUT0442
3V6)*G2                                                               COUT0443
 TM$(1,4)=TM$(1,4)                                                    COUT0444
1-(24.D0*U1-16.D0*U2+4.D0*U3-48.D0*U4-16.D0*U5-32.D0*U6)*F2           COUT0445
2               -(24.D0*V1-16.D0*V2+4.D0*V3-48.D0*V4-16.D0*V5-32.D0*V6COUT0446
3)*G2                         -(24.D0*U1-32.D0*U2-16.D0*U3-48.D0*U4+4.D0COUT0447
4*U5-16.D0*U6)*F3                        -(24.D0*V1-32.D0*V2-16.D0*V3-4COUT0448
58.D0*V4+4.D0*V5-16.D0*V6)*G3                                         COUT0449
 TM$(2,4)=TM$(2,4)                                                    COUT0450
1-(-16.D0*U1+128.D0*U2-16.D0*U3+128.D0*U4-16.D0*U5+128.D0*U6)*F2      COUT0451
2               -(-16.D0*V1+128.D0*V2-16.D0*V3+128.D0*V4-16.D0*V5+128.COUT0452
3D0*V6)*G2                    -(-32.D0*U1+384.D0*U2+48.D0*U3+192.D0*U4-4COUT0453
48.D0*U5+128.D0*U6)*F3                   -(-32.D0*V1+384.D0*V2+48.D0*V3COUT0454
5+192.D0*V4-48.D0*V5+128.D0*V6)*G3                                    COUT0455
 TM$(3,4)=TM$(3,4)                                                    COUT0456
1-(4.D0*U1-16.D0*U2+24.D0*U3-32.D0*U4+16.D0*U5-48.D0*U6)*F2           COUT0457
2               -(4.D0*V1-16.D0*V2+24.D0*V3-32.D0*V4+16.D0*V5-48.D0*V6COUT0458
3)*G2                         -(-16.D0*U1+48.D0*U2+120.D0*U3+48.D0*U4-16COUT0459
4.D0*U5-16.D0*U6)*F3                     -(-16.D0*V1+48.D0*V2+120.D0*V3COUT0460
5+48.D0*V4-16.D0*V5-16.D0*V6)*G3                                      COUT0461
 TM$(4,4)=TM$(4,4)                                                    COUT0462
1-(-48.D0*U1+128.D0*U2-32.D0*U3+384.D0*U4+48.D0*U5+192.D0*U6)*F2      COUT0463
2               -(-48.D0*V1+128.D0*V2-32.D0*V3+384.D0*V4+48.D0*V5+192.COUT0464
3D0*V6)*G2                    -(-48.D0*U1+192.D0*U2+48.D0*U3+384.D0*U4-3COUT0465
42.D0*U5+128.D0*U6)*F3                   -(-48.D0*V1+192.D0*V2+48.D0*V3COUT0466
5+384.D0*V4-32.D0*V5+128.D0*V6)*G3                                    COUT0467
 TM$(5,4)=TM$(5,4)                                                    COUT0468
```

```
     1-(-16.D0*U1-16.D0*U2-16.D0*U3+48.D0*U4+120.D0*U5+48.D0*U6)*F2     COUT0469
     2          -(-16.D0*V1-16.D0*V2-16.D0*V3+48.D0*V4+120.D0*V5+48.D0COUT0470
     3*V6)*G2                 -(4.D0*U1-48.D0*U2-16.D0*U3-32.D0*U4+24.D0COUT0471
     4*U5-16.D0*U6)*F3                  -(4.D0*V1-48.D0*V2-16.D0*V3-32COUT0472
     5.D0*V4+24.D0*V5-16.D0*V6)*G3                                      COUT0473
      TM$(6,4)=TM$(6,4)                                                 COUT0474
     1-(-32.D0*U1+128.D0*U2-48.D0*U3+192.D0*U4+48.D0*U5+384.D0*U6)*F2   COUT0475
     2          -(-32.D0*V1+128.D0*V2-48.D0*V3+192.D0*V4+48.D0*V5+384.COUT0476
     3D0*V6)*G2               -(-16.D0*U1+128.D0*U2-16.D0*U3+128.D0*U4-1COUT0477
     46.D0*U5+128.D0*U6)*F3             -(-16.D0*V1+128.D0*V2-16.D0*V3COUT0478
     5+128.D0*V4-16.D0*V5+128.D0*V6)*G3                                 COUT0479
      TM$(1,5)=TM$(1,5)                                                 COUT0480
     1-(-18.D0*U1-16.D0*U2+11.D0*U3-20.D0*U4-9.D0*U5-32.D0*U6)*F3       COUT0481
     2          -(-18.D0*V1-16.D0*V2+11.D0*V3-20.D0*V4-9.D0*V5-32.D0*VCOUT0482
     36)*G3                                                             COUT0483
      TM$(2,5)=TM$(2,5)                                                 COUT0484
     1-(-16.D0*U1-96.D0*U2-16.D0*U3+16.D0*U4+12.D0*U5+16.D0*U6)*F3      COUT0485
     2          -(-16.D0*V1-96.D0*V2-16.D0*V3+16.D0*V4+12.D0*V5+16.D0*COUT0486
     3V6)*G3                                                            COUT0487
      TM$(3,5)=TM$(3,5)                                                 COUT0488
     1-(11.D0*U1-16.D0*U2-18.D0*U3-32.D0*U4-9.D0*U5-20.D0*U6)*F3        COUT0489
     2          -(11.D0*V1-16.D0*V2-18.D0*V3-32.D0*V4-9.D0*V5-20.D0*V6COUT0490
     3)*G3                                                              COUT0491
      TM$(4,5)=TM$(4,5)                                                 COUT0492
     1-(-20.D0*U1+16.D0*U2-32.D0*U3+160.D0*U4+48.D0*U5+80.D0*U6)*F3     COUT0493
     2          -(-20.D0*V1+16.D0*V2-32.D0*V3+160.D0*V4+48.D0*V5+80.D0COUT0494
     3*V6)*G3                                                           COUT0495
      TM$(5,5)=TM$(5,5)                                                 COUT0496
     1-(-9.D0*U1+12.D0*U2-9.D0*U3+48.D0*U4+78.D0*U5+48.D0*U6)*F3        COUT0497
     2          -(-9.D0*V1+12.D0*V2-9.D0*V3+48.D0*V4+78.D0*V5+48.D0*V6COUT0498
     3)*G3                                                              COUT0499
      TM$(6,5)=TM$(6,5)                                                 COUT0500
     1-(-32.D0*U1+16.D0*U2-20.D0*U3+80.D0*U4+48.D0*U5+160.D0*U6)*F3     COUT0501
     2          -(-32.D0*V1+16.D0*V2-20.D0*V3+80.D0*V4+48.D0*V5+160.D0COUT0502
     3*V6)*G3                                                           COUT0503
      TM$(1,6)=TM$(1,6)                                                 COUT0504
     1-(24.D0*U1-16.D0*U2+4.D0*U3-48.D0*U4-16.D0*U5-32.D0*U6)*F1        COUT0505
     2          -(24.D0*V1-16.D0*V2+4.D0*V3-48.D0*V4-16.D0*V5-32.D0*V6COUT0506
     3)*G1                    -(120.D0*U1+48.D0*U2-16.D0*U3-16.D0*U4-16.COUT0507
     4D0*U5+48.D0*U6)*F3                -(120.D0*V1+48.D0*V2-16.D0*V3-COUT0508
     516.D0*V4-16.D0*V5+48.D0*V6)*G3                                    COUT0509
      TM$(2,6)=TM$(2,6)                                                 COUT0510
     1-(-16.D0*U1+128.D0*U2-16.D0*U3+128.D0*U4-16.D0*U5+128.D0*U6)*F1   COUT0511
     2          -(-16.D0*V1+128.D0*V2-16.D0*V3+128.D0*V4-16.D0*V5+128.COUT0512
     3D0*V6)*G1               -(48.D0*U1+384.D0*U2-32.D0*U3+128.D0*U4-48COUT0513
     4.D0*U5+192.D0*U6)*F3              -(48.D0*V1+384.D0*V2-32.D0*V3+COUT0514
     5128.D0*V4-48.D0*V5+192.D0*V6)*G3                                  COUT0515
      TM$(3,6)=TM$(3,6)                                                 COUT0516
```

```
1-(4.D0*U1-16.D0*U2+24.D0*U3-32.D0*U4-16.D0*U5-48.D0*U6)*F1        COUT0517
2           -(4.D0*V1-16.D0*V2+24.D0*V3-32.D0*V4-16.D0*V5-48.D0*V6COUT0518
3)*G1                   -(-16.D0*U1-32.D0*U2+24.D0*U3-16.D0*U4+4.DCOUT0519
40*U5-48.D0*U6)*F3                  -(-16.D0*V1-32.D0*V2+24.D0*V3-COUT0520
516.D0*V4+4.D0*V5-48.D0*V6)*G3                                     COUT0521
 TM$(4,6)=TM$(4,6)                                                 COUT0522
1-(-48.D0*U1+128.D0*U2-32.D0*U3+384.D0*U4+48.D0*U5+192.D0*U6)*F1   COUT0523
2           -(-48.D0*V1+128.D0*V2-32.D0*V3+384.D0*V4+48.D0*V5+192.COUT0524
3D0*V6)*G1              -(-16.D0*U1+128.D0*U2-16.D0*U3+128.D0*U4-1COUT0525
46.D0*U5+128.D0*U6)*F3              -(-16.D0*V1+128.D0*V2-16.D0*V3COUT0526
5+128.D0*V4-16.D0*V5+128.D0*V6)*G3                                 COUT0527
 TM$(5,6)=TM$(5,6)                                                 COUT0528
1-(-16.D0*U1-16.D0*U2-16.D0*U3+48.D0*U4+120.D0*U5+48.D0*U6)*F1     COUT0529
2           -(-16.D0*V1-16.D0*V2-16.D0*V3+48.D0*V4+120.D0*V5+48.D0COUT0530
3*V6)*G1                -(-16.D0*U1-48.D0*U2+4.D0*U3-16.D0*U4+24.DCOUT0531
40*U5-32.D0*U6)*F3                  -(-16.D0*V1-48.D0*V2+4.D0*V3-1COUT0532
56.D0*V4+24.D0*V5-32.D0*V6)*G3                                     COUT0533
 TM$(6,6)=TM$(6,6)                                                 COUT0534
1-(-32.D0*U1+128.D0*U2-48.D0*U3+192.D0*U4+48.D0*U5+384.D0*U6)*F1   COUT0535
2           -(-32.D0*V1+128.D0*V2-48.D0*V3+192.D0*V4+48.D0*V5+384.COUT0536
3D0*V6)*G1              -(48.D0*U1+192.D0*U2-48.D0*U3+128.D0*U4-32COUT0537
4.D0*U5+384.D0*U6)*F3               -(48.D0*V1+192.D0*V2-48.D0*V3+COUT0538
5128.D0*V4-32.D0*V5+384.D0*V6)*G3                                  COUT0539
C                                                                  COUT0540
C THIS ENDS ADDITION OF NONLINEAR TERMS TO THE LOCAL ARRAY          COUT0541
C                                                                  COUT0542
 TM$(7,7) = TM$(1,1)                                               COUT0543
 TM$(7,8) = TM$(1,2)                                               COUT0544
 TM$(7,9) = TM$(1,3)                                               COUT0545
 TM$(7,10) = TM$(1,4)                                              COUT0546
 TM$(7,11) = TM$(1,5)                                              COUT0547
 TM$(7,12) = TM$(1,6)                                              COUT0548
 TM$(8,7) = TM$(2,1)                                               COUT0549
 TM$(8,8) = TM$(2,2)                                               COUT0550
 TM$(8,9) = TM$(2,3)                                               COUT0551
 TM$(8,10) = TM$(2,4)                                              COUT0552
 TM$(8,11) = TM$(2,5)                                              COUT0553
 TM$(8,12) = TM$(2,6)                                              COUT0554
 TM$(9,7) = TM$(3,1)                                               COUT0555
 TM$(9,8) = TM$(3,2)                                               COUT0556
 TM$(9,9) = TM$(3,3)                                               COUT0557
 TM$(9,10) = TM$(3,4)                                              COUT0558
 TM$(9,11) = TM$(3,5)                                              COUT0559
 TM$(9,12) = TM$(3,6)                                              COUT0560
 TM$(10,7) = TM$(4,1)                                              COUT0561
 TM$(10,8) = TM$(4,2)                                              COUT0562
 TM$(10,9) = TM$(4,3)                                              COUT0563
 TM$(10,10) = TM$(4,4)                                             COUT0564
```

```
      TM$(10,11) = TM$(4,5)                                          COUT0565
      TM$(10,12) = TM$(4,6)                                          COUT0566
      TM$(11,7) = TM$(5,1)                                           COUT0567
      TM$(11,8) = TM$(5,2)                                           COUT0568
      TM$(11,9) = TM$(5,3)                                           COUT0569
      TM$(11,10) = TM$(5,4)                                          COUT0570
      TM$(11,11) = TM$(5,5)                                          COUT0571
      TM$(11,12) = TM$(5,6)                                          COUT0572
      TM$(12,7) = TM$(6,1)                                           COUT0573
      TM$(12,8) = TM$(6,2)                                           COUT0574
      TM$(12,9) = TM$(6,3)                                           COUT0575
      TM$(12,10) = TM$(6,4)                                          COUT0576
      TM$(12,11) = TM$(6,5)                                          COUT0577
      TM$(12,12) = TM$(6,6)                                          COUT0578
      TM$(13,1) = D1                                                 COUT0579
      TM$(1,13) = TM$(13,1)                                          COUT0580
      TM$(14,1) = 0.D0                                               COUT0581
      TM$(1,14) = TM$(14,1)                                          COUT0582
      TM$(15,1) = 0.D0                                               COUT0583
      TM$(1,15) = TM$(15,1)                                          COUT0584
      TM$(13,2) = D1+2.D0*D2                                         COUT0585
      TM$(2,13) = TM$(13,2)                                          COUT0586
      TM$(14,2) = 2.D0*D1+D2                                         COUT0587
      TM$(2,14) = TM$(14,2)                                          COUT0588
      TM$(15,2) = D1+D2                                              COUT0589
      TM$(2,15) = TM$(15,2)                                          COUT0590
      TM$(13,3) = 0.D0                                               COUT0591
      TM$(3,13) = TM$(13,3)                                          COUT0592
      TM$(14,3) = D2                                                 COUT0593
      TM$(3,14) = TM$(14,3)                                          COUT0594
      TM$(15,3) = 0.D0                                               COUT0595
      TM$(3,15) = TM$(15,3)                                          COUT0596
      TM$(13,4) = D2+D3                                              COUT0597
      TM$(4,13) = TM$(13,4)                                          COUT0598
      TM$(14,4) = D2+2.D0*D3                                         COUT0599
      TM$(4,14) = TM$(14,4)                                          COUT0600
      TM$(15,4) = 2.D0*D2+D3                                         COUT0601
      TM$(4,15) = TM$(15,4)                                          COUT0602
      TM$(13,5) = 0.D0                                               COUT0603
      TM$(5,13) = TM$(13,5)                                          COUT0604
      TM$(14,5) = 0.D0                                               COUT0605
      TM$(5,14) = TM$(14,5)                                          COUT0606
      TM$(15,5) = D3                                                 COUT0607
      TM$(5,15) = TM$(15,5)                                          COUT0608
      TM$(13,6) = D1+2.D0*D3                                         COUT0609
      TM$(6,13) = TM$(13,6)                                          COUT0610
      TM$(14,6) = D1+D3                                              COUT0611
      TM$(6,14) = TM$(14,6)                                          COUT0612
```

```
      TM$(15,6) = 2.D0*D1+D3                                             COUT0613
      TM$(6,15) = TM$(15,6)                                              COUT0614
      TM$(13,7) = E1                                                     COUT0615
      TM$(7,13) = TM$(13,7)                                              COUT0616
      TM$(14,7) = 0.D0                                                   COUT0617
      TM$(7,14) = TM$(14,7)                                              COUT0618
      TM$(15,7) = 0.D0                                                   COUT0619
      TM$(7,15) = TM$(15,7)                                              COUT0620
      TM$(13,8) = E1+2.D0*E2                                             COUT0621
      TM$(8,13) = TM$(13,8)                                              COUT0622
      TM$(14,8) = 2.D0*E1+E2                                             COUT0623
      TM$(8,14) = TM$(14,8)                                              COUT0624
      TM$(15,8) = E1+E2                                                  COUT0625
      TM$(8,15) = TM$(15,8)                                              COUT0626
      TM$(13,9) = 0.D0                                                   COUT0627
      TM$(9,13) = TM$(13,9)                                              COUT0628
      TM$(14,9) = E2                                                     COUT0629
      TM$(9,14) = TM$(14,9)                                              COUT0630
      TM$(15,9) = 0.D0                                                   COUT0631
      TM$(9,15) = TM$(15,9)                                              COUT0632
      TM$(13,10) = E2+E3                                                 COUT0633
      TM$(10,13) = TM$(13,10)                                            COUT0634
      TM$(14,10) = E2+2.D0*E3                                            COUT0635
      TM$(10,14) = TM$(14,10)                                            COUT0636
      TM$(15,10) = E2+E2+E3                                              COUT0637
      TM$(10,15) = TM$(15,10)                                            COUT0638
      TM$(13,11) = 0.D0                                                  COUT0639
      TM$(11,13) = TM$(13,11)                                            COUT0640
      TM$(14,11) = 0.D0                                                  COUT0641
      TM$(11,14) = TM$(14,11)                                            COUT0642
      TM$(15,11) = E3                                                    COUT0643
      TM$(11,15) = TM$(15,11)                                            COUT0644
      TM$(13,12) = E1+E3+E3                                              COUT0645
      TM$(12,13) = TM$(13,12)                                            COUT0646
      TM$(14,12) = E1+E3                                                 COUT0647
      TM$(12,14) = TM$(14,12)                                            COUT0648
      TM$(15,12) = 2.D0*E1+E3                                            COUT0649
      TM$(12,15) = TM$(15,12)                                            COUT0650
  185 CONTINUE                                                           COUT0651
      N(1)=N1                                                            COUT0652
      N(2)=N2                                                            COUT0653
      N(3)=N3                                                            COUT0654
      N(4)=N4                                                            COUT0655
      N(5)=N5                                                            COUT0656
      N(6)=N6                                                            COUT0657
      N(7)=N7                                                            COUT0658
      N(8)=N8                                                            COUT0659
      N(9)=N9                                                            COUT0660
```

```
      N(10)=N10                                                         COUT0661
      N(11)=N11                                                         COUT0662
      N(12)=N12                                                         COUT0663
      N(13)=N13                                                         COUT0664
      N(14)=N14                                                         COUT0665
      N(15)=N15                                                         COUT0666
C                                                                       COUT0667
      DO 3100 I$=1,15                                                   COUT0668
      I = N(I$)                                                         COUT0669
C                                                                       COUT0670
      DO 3100 J$=1,15                                                   COUT0671
      J = N(J$)                                                         COUT0672
      TM(I,J) = TM(I,J)+TM$(I$,J$)                                      COUT0673
 3100 CONTINUE                                                          COUT0674
C                                                                       COUT0675
 3200 CONTINUE                                                          COUT0676
C                                                                       COUT0677
C                                                                       COUT0678
      DO 3300 I=1,NNQXY                                                 COUT0679
      RHS(NQS(I)) = RHS(NQS(I))+Q(NQS(I))                               COUT0680
      RHS(NQS(I)+NN) = RHS(NQS(I)+NN)+Q(NQS(I)+NN)                      COUT0681
 3300 CONTINUE                                                          COUT0682
C                                                                       COUT0683
 3400 CONTINUE                                                          COUT0684
C                                                                       COUT0685
C  INSERT SYSTEM BOUNDARY CONDITIONS                                    COUT0686
C                                                                       COUT0687
      DO 3500 I=1,MM                                                    COUT0688
C                                                                       COUT0689
      DO 3500 J=1,NTOTVP                                                COUT0690
      JX = NVIS(J)                                                      COUT0691
      RHS(I) = RHS(I)-TM(I,JX)*T(JX)                                    COUT0692
      TM(I,JX) = 0.D0                                                   COUT0693
      TM(JX,I) = 0.D0                                                   COUT0694
      TM(JX,JX) = 1.D0                                                  COUT0695
      RHS(JX) = T(JX)                                                   COUT0696
 3500 CONTINUE                                                          COUT0697
C                                                                       COUT0698
      M = 1                                                             COUT0699
      ND = 82                                                           COUT0700
      IA = 82                                                           COUT0701
      IDGT = 0                                                          COUT0702
      CALL LEQT2F (TM,M,ND,IA,RHS,IDGT,WKAREA,IER)                      COUT0703
      WRITE (NWRITE,5700)                                               COUT0704
      DO 322 J=1,MM                                                     COUT0705
      TDIFF=DABS(T1(J)-T(J))                                            COUT0706
      EPSLN=1.D-06                                                      COUT0707
      IF(TDIFF-EPSLN) 322,324,324                                       COUT0708
```

```
  322 CONTINUE
  324 CONTINUE
C
      DO 3600 I=1,MM
      T(I) = RHS(I)
      T1(I)=T(I)
      WRITE (NWRITE,5800) I,T(I)
 3600 CONTINUE
      IF(J.NE.MM) GO TO 2840
      WRITE(NWRITE,1091)
C
 1090 FORMAT(////,5X,'NODAL VARIABLE IS THE U-VELOCITY AT NODES 1 - 35;'
     1,//,5X,'THE V-VELOCITY AT NODES 36 - 70;',
     2//,5X,'AND THE PRESSURES AT NODES 71 - 82.',/////)
 1091 FORMAT(1H1)
 1092 FORMAT(////,5X,'THE FIRST SEQUENCE OF 82 NODAL VARIABLES',
     1//,5X,'REPRESENTS A LINEAR, STEADY STATE SYSTEM;',
     2//,5X,'WHILE THE SECOND SET OF THE 82 VALUES CORRESPONDS',
     3//,5X,'TO A NONLINEAR ANALYSIS OF THE SOLUTION DOMAIN.',/////)
 1093 FORMAT(////,5X,'A PRESSURE GRADIENT IN THE HORIZONTAL SHEAR'
     1,//,5X,'DIRECTION OF -3 UNITS IN MAGNITUDE HAS BEEN ADDED'
     2,//,5X,'TO PRODUCE A CURVED VELOCITY PROFILE.',/////)
 1094 FORMAT(1H1,15X,'THIS IS A 2-D NONLINEAR COUETTE FLOW PROBLEM',//)
 3700 FORMAT (5X,'IBAND=',I3,//,5X,'NEQ =',I3,//)
 3800 FORMAT (3I10)
 3900 FORMAT (6X,A4,I10,2F10.0)
 4000 FORMAT (7I10)
 4100 FORMAT (6X,A4,I10,F10.0)
 4300 FORMAT (5X,'NO. OF NODES=',I3,//,5X,'NO. OF ELEMENTS=',I3,//,
     1 5X,'NO. OF CORNER NODES=',I3,///)
 4400 FORMAT (5X,'NNVELS=',I3,//)
 4500 FORMAT (5X,'NNQXY=',I3,//)
 4600 FORMAT (5X,'NNPS=',I3,//)
 4700 FORMAT (5X,'SUMMARY OF NODAL COORDINATES',//,
     1 7X,'I',12X'X(I)',13X,'Y(I)',//)
 4800 FORMAT (5X,I3,2(7X,F10.3))
 4900 FORMAT(//,5X,'LISTING OF SYSTEM TOPOLOGY',//,5X,
     1 'ELEMENT NUMBER',20X,'NODE NUMBERS',//)
 5000 FORMAT (5X,I3,10X,6(5X,I3))
 5100 FORMAT (//,7X,'NODES WHERE VELOCITIES ARE SPECIFIED',
     1 //,8X,'I',5X,'NODE',5X,'U VELOCITY',5X,'V VELOCITY',//)
 5200 FORMAT (2X,2(4X,I3),3X,F12.3,3X,F12.3)
 5300 FORMAT (//,5X,'NODES WHERE QX AND QY ARE SPECIFIED',
     1          //,2X,'      I      NODE',1X,'          QX',10X,'    QY',//)
 5400 FORMAT (//,5X,'NODES WHERE PRESSURE IS SPECIFIED',
     1             //,5X,'I',5X,'NODE',15X,'PRESSURE',//)
 5600 FORMAT (7X,I3,3X,I3,10X,F12.3)
 5700 FORMAT (///,5X,'NODE NO.',6X,'NODE VARIABLES',//)
```

```
5800 FORMAT (9X,I3,5X,D17.10,/)                                         COUT0757
     STOP                                                               COUT0758
     END                                                                COUT0759
```

STEADY STATE FLUID MECHANICS PROBLEM

```
      IMPLICIT REAL*8(A-H,O-Z,$)                                       FLSS0110
      CATA NREAD/5/                                                    FLSS0120
                                                                       FLSS0130
THE U VELOCITY IS IN THE FIRST NN POSITIONS OF X(I)                    FLSS0140
THE V VELOCITY IS IN THE SECOND NN POSITIONS OF X I.E. X(I+NN)         FLSS0150
THE P PRESSURE IS IN THE NN+NN+I POSITIONS OF X I.E. X(I+NN+NN)        FLSS0160
THE T TEMPERATURE IS IN THE NN+NN+NNCN+I PPOSITIONS OF X I.E.          FLSS0170
X(I+NN+NN+NNCN)                                                        FLSS0180
THERE ARE NNCN PRESSURE NODES (NNCN=NUMBER OF CORNER NODES)            FLSS0190
                                                                       FLSS0200
TM MUST BE DIMENSIONED 3*NN+NNCN X 3*NN+NNCN                           FLSS0210
                                                                       FLSS0220
                                                                       FLSS0230
      DATA NWRITE/6/                                                   FLSS0240
      DATA STOP/'STOP'/                                                FLSS0250
      DIMENSION XC(125),YC(125),NODE(125,6),NVS(125),NCN(125)          FLSS0260
      DIMENSION X(117),NVIS(117),NCP(125),NPS(125),Q(117)              FLSS0270
      DIMENSION NQS(125),T1(117)                                       FLSS0280
      DIMENSION TM$(21,21),N(21),NQIS(117)                             FLSS0290
      DIMENSION RP$(6),ZP$(6),WKAREA(15000)                            FLSS0300
      DIMENSION XC$(3),YC$(3),RHS(117),TM(117,117)                     FLSS0310
                                                                       FLSS0320
SPECIFY WHETHER TWO DIMENSIONAL, INCLUDING NON-LINEAR TERMS,           FLSS0330
(NCASE=1) OR AXISYMMETRIC (NCASE=2)                                    FLSS0340
                                                                       FLSS0350
      READ(NREAD,500)NCASE                                             FLSS0360
      WRITE(NWRITE,600)
      IF(NCASE.EQ.1)GO TO 5                                            FLSS0370
      WRITE(NWRITE,2015)                                               FLSS0380
      GC TO 6                                                          FLSS0390
    5 WRITE(NWRITE,2020)                                               FLSS0400
    6 CONTINUE                                                         FLSS0410
                                                                       FLSS0420
THE FIRST PART OF THE PROGRAM CAN BE CONSIDERED AS AN INPUT ROUTINE    FLSS0430
IN WHICH LINES 540 TO 2970 ARE INPUT VERIFICATION OF ALL DATA.         FLSS0440
SUCH A SECTION WOULD BE PART OF ANY FINITE ELEMENT PROGRAM.            FLSS0460
```

```
                                                                           FLSS0470
                                                                           FLSS0480
  READ IN NUMBER OF NODES AND ELEMENTS AND NO. OF CCRNER NODES             FLSS0490
        READ(NREAD,1005)NN,NE,NNCN                                         FLSS0500
                                                                           FLSS0510
  INITIALIZE ALL PARAMETERS                                                FLSS0520
                                                                           FLSS0530
        MM=2*NN+NNCN                                                       FLSS0540
        MMM=3*NN+NNCN                                                      FLSS0550
        DO 50 I=1,MMM                                                      FLSS0560
        XC(I)=0.DO                                                         FLSS0570
        YC(I)=0.DO                                                         FLSS0580
        NVS(I)=0                                                           FLSS0590
        NCN(I)=0                                                           FLSS0600
        NCP(I)=0                                                           FLSS0610
        NPS(I)=0                                                           FLSS0620
        NCS(I)=0                                                           FLSS0630
        DO 50 J=1,6                                                        FLSS0640
        NODE(I,J)=0                                                        FLSS0650
     50 CCNTINUE                                                           FLSS0660
        DO 51 I=1,MMM                                                      FLSS0670
        T1(I)=0.DO                                                         FLSS0680
        X(I)=0.DO                                                          FLSS0690
        NVIS(I)=0                                                          FLSS0700
        Q(I)=0.DO                                                          FLSS0710
        NCIS(I)=0                                                          FLSS0720
        RHS(I)=0.DO                                                        FLSS0730
        DO 51 J=1,MMM                                                      FLSS0740
        TM(I,J)=0.DO                                                       FLSS0750
     51 CCNTINUE                                                           FLSS0760
        DO 52 I=1,21                                                       FLSS0770
        N(I)=0                                                             FLSS0780
        DO 52 J=1,21                                                       FLSS0790
        TM$(I,J)=0.DO                                                      FLSS0800
     52 CONTINUE                                                           FLSS0810
        DO 53 I=1,6                                                        FLSS0820
        RP$(I)=0.DO                                                        FLSS0830
        ZP$(I)=0.DO                                                        FLSS0840
     53 CCNTINUE                                                           FLSS0850
        DO 54 I=1,3                                                        FLSS0860
        XC$(I)=0.DO                                                        FLSS0870
        YC$(I)=0.DO                                                        FLSS0880
     54 CONTINUE                                                           FLSS0890
                                                                           FLSS0900
  READ NODE NUMBER AND COORDINATES                                         FLSS0910
                                                                           FLSS0920
        DO 100 J=1,NN                                                      FLSS0930
```

```
      READ(NREAD,1006)WORD,I,XC(I),YC(I)
      IF(WORD.EQ.STOP) GO TO 101
      NCN(J)=I
  100 CONTINUE
  101 NNCN=J-1

THE ARRAY NCP(J) GENERATES THE GLOBAL PRESSURE INDICES (P1,P2.ETC.)
THUS PRESSURE NODES ARE LABELED AS CORNER NODES AND ARE INPUTED
WHEN ONE INPUTS A GLOBAL CORNER NODE FOR J

      DO 107 J=1,NNCN
      NCP(NCN(J))=J+NN+NN
  107 CONTINUE

READ SYSTEM TOPOLOGY (ELEMENT NO. AND NODE NUMBERS IN
COUNTERCLOCKWISE FASHION STARTING AT THE UPPER LEFT
HAND CORNER NODE).

      DO 105 I=1,NE
      READ(NREAD,1010)J,NODE(J,1),NODE(J,2),NODE(J,3),
     1NODE(J,4),NODE(J,5),NODE(J,6)
  105 CONTINUE
      MAXDIF=0
      DO 108 I=1,NE
      DO 108 J=1,6
      DO 108 K=1,6
      LL=IABS(NODE(I,J)-NODE(I,K))
      IF(LL.GT.MAXDIF) MAXDIF=LL
      IBAND=2*(MAXDIF+1)
      NEQ=3*NN+NNCN
  108 CONTINUE
      WRITE(NWRITE,1017)IBAND,NEQ
 1017 FORMAT(5X,'IBAND=',I3,//,5X,'NEQ=',I3,//)

READ NODES WHERE BOTH U AND V VELOCITY IS SPECIFIED

      DO 110 I=1,MM
      READ(NREAD,1006)WORD,NVELS,VELU,VELV
      IF(WORD.EQ.STOP) GO TO 111
      NVS(I)=NVELS
      X(NVS(I))=VELU
      X(NVS(I)+NN)=VELV
  110 CONTINUE

COUNT NODES HAVING SPECIFIED VELOCITIES

  111 NNVELS=I-1
```

```
READ QX AND QY VALUES AT INTERNAL NODES

      DO 125 I=1,NN
      READ(NREAD,1006)WORD,NQXY,QXNS,QYNS
      IF(WORD.EQ.STOP) GO TO 126
      NQS(I)=NQXY
      Q(NQS(I))=QXNS
      Q(NQS(I)+NN)=QYNS
  125 CONTINUE

COUNT NODES HAVING SPECIFIED QX AND QY

  126 NNQXY=I-1

READ NODE NUMBER AND PRESSURE WHERE SPECIFIED

      DO 130 I=1,NN
      READ(NREAD,1025)WORD,NP,PNP
      IF(WORD.EQ.STOP)GO TO 135
      NPS(I)=NP
      X(NCP(NPS(I)))=PNP
  130 CONTINUE

COUNT BOUNDARY NODES WHERE PRESSURE IS SPECIFIED

  135 NNPS=I-1

READ NODE NUMBER AND TEMPERATURE WHERE SPECIFIED

      DO 140 I=1,MM
      READ(NREAD,1025) WORD,NTEMP,TNT
      IF(WORD.EQ.STOP) GO TO  145
      NVS(I+ NNVELS)=NTEMP
      X(NVS(I+NNVELS)+MM)=TNT
  140 CONTINUE

COUNT NODES HAVING SPECIFIED TEMPERATURES

  145 NNTS=I-1

READ NODE NUMBERS AND QZC WHERE SPECIFIED

      DO 141 I=1,MM
      READ(NREAD,1025) WORD,NQZC,QZCNS
      IF(WORD.EQ.STOP) GO TO 146
      NQS(NNQXY+I)=NQZC
      Q(NQS(NNQXY+I)+2*NN)=QZCNS
  141 CONTINUE
```

```
COUNT NODES WHERE QZC IS SPECIFIED

  146 NNQZC=I-1

READ NODE NUMBERS AND HEAT FLUX QZ WHERE SPECIFIED

      DO 142 I=1,MM
      READ(NREAD,1025) WORD,NQZ,QZNS
      IF(WORD.EQ.STOP) GO TO  147
      NQS(NNQXY+NNQZC+I)=NQZ
      Q(NQS(NNQXY+NNQZC+I)+MM)=QZNS
  142 CONTINUE

COUNT NODES WHERE HEAT FLUX QZ IS SPECIFIED

  147 NNQZ=I-1

NQIS IS A LIST OF INDICES OF KNOWN QX,QY,QZC AND QZ

      DO 1140 I=1,NNQXY
      NQIS(I)=NQS(I)
      NQIS(I+NNQXY)=NQS(I)+NN
 1140 CONTINUE
      DO 1141 I=1,NNQZC
      NQIS(2*NNQXY+I)=NQS(NNQXY+I)+2*NN
 1141 CONTINUE
      DO 1145 I=1,NNQZ
      NQIS(2*NNQXY+NNQZC+I)=NQS(NNQXY+NNQZC+I)+MM
 1145 CONTINUE

NVIS IS A LIST OF KNOWN VELOCITY, PRESSURE, AND TEMPERATURE INDICES

      DO 1150 I=1,NNVELS
      NVIS(I)=NVS(I)
      NVIS(I+NNVELS)=NVS(I)+NN
 1150 CONTINUE
      DO 1155 J=1,NNPS
      NVIS(2*NNVELS+J)=NCP(NPS(J))
 1155 CONTINUE
      DO 1160 K=1,NNTS
      NVIS(2*NNVELS+NNPS+K)=NVS(K+NNVELS)+MM
 1160 CONTINUE

NNHC=NUMBER OF NODES WHERE HEAT TRANSFER COEFFICIENT IS SPECIFIED

      NNHC=0
```

```
NTOTC=TOTAL NUMBER OF KNOWN QX,QY,QZC, AND QZ                       FLSS2380
                                                                    FLSS2390
      NTOTQ=2*NNQXY+NNQZC+NNQZ                                      FLSS2400
                                                                    FLSS2410
NTOTVP=TOTAL NUMBER OF KNOWN VELOCITIES, PRESSURES, AND TEMPERATURES  FLSS2420
                                                                    FLSS2430
      NTOTVP= 2*NNVELS+NNPS+NNTS                                    FLSS2440
                                                                    FLSS2450
PRINT ALL INPUT DATA                                                FLSS2460
                                                                    FLSS2470
      WRITE(NWRITE,1035)NN,NE,NNCN                                  FLSS2480
      WRITE(NWRITE,1036)NNVELS                                      FLSS2490
      WRITE(NWRITE,1037)NNQXY                                       FLSS2500
      WRITE(NWRITE,1038)NNPS                                        FLSS2510
      WRITE(NWRITE,1039)NNTS                                        FLSS2520
      WRITE(NWRITE,1034)NNQZC                                       FLSS2530
      WRITE(NWRITE,1040)NNQZ                                        FLSS2540
      WRITE(NWRITE,1041)                                            FLSS2550
      DO 150 I=1,NNCN                                               FLSS2560
      WRITE(NWRITE,1045)NCN(I),XC(NCN(I)),YC(NCN(I))                FLSS2570
  150 CONTINUE                                                      FLSS2580
      WRITE(NWRITE,1050)                                            FLSS2590
      DO 155 I=1,NE                                                 FLSS2600
      WRITE(NWRITE,1055)I,NODE(I,1),NODE(I,2),NODE(I,3),            FLSS2610
     1NODE(I,4),NODE(I,5),NODE(I,6)                                 FLSS2620
  155 CONTINUE                                                      FLSS2630
      WRITE(NWRITE,1060)                                            FLSS2640
      DO 160 I=1,NNVELS                                             FLSS2650
      WRITE(NWRITE,1065)I,NVS(I),X(NVS(I)),X(NVS(I)+NN)             FLSS2660
  160 CONTINUE                                                      FLSS2670
      WRITE(NWRITE,1070)                                            FLSS2680
      DO 165 I=1,NNQXY                                              FLSS2690
      WRITE(NWRITE,1065)I,NQS(I),Q(NQS(I)),Q(NQS(I)+NN)             FLSS2700
  165 CONTINUE                                                      FLSS2710
      WRITE(NWRITE,1080)                                            FLSS2720
      DO 170 I=1,NNPS                                               FLSS2730
      WRITE(NWRITE,1085)I,NPS(I),X(NCP(NPS(I)))                     FLSS2740
  170 CONTINUE                                                      FLSS2750
      WRITE(NWRITE,1081)                                            FLSS2760
      DO 171 I=1,NNTS                                               FLSS2770
      WRITE(NWRITE,1085)I,NVS(I+NNVELS),X(NVS(I+NNVELS)+MM)         FLSS2780
  171 CONTINUE                                                      FLSS2790
      WRITE(NWRITE,1083)                                            FLSS2800
      DO 173 I=1,NNQZC                                              FLSS2810
      WRITE(NWRITE,1085)I,NQS(I+NNQXY),Q(NQS(I+NNCXY)+2*NN)         FLSS2820
  173 CONTINUE                                                      FLSS2830
      WRITE(NWRITE,1082)                                            FLSS2840
      DO 172 I=1,NNQZ                                               FLSS2850
```

```
      WRITE(NWRITE,1085)I,NQS(I+NNQXY+NNQZC),Q(NQS(I+NNQXY+NNQZC)+MM)   FLSS2860
  172 CONTINUE                                                           FLSS2870
      WRITE(NWRITE,1090)
      WRITE(NWRITE,2055)                                                 FLSS8320
      WRITE(NWRITE,1091)
  177 CONTINUE                                                           FLSS2880
      DO 178 I=1,MMM                                                     FLSS2890
      RHS(I)=0.D0                                                        FLSS2900
      DO 178 J=1,MMM                                                     FLSS2910
      TM(I,J)=0.D0                                                       FLSS2920
  178 CONTINUE                                                           FLSS2930
      DO 181 I=1,21                                                      FLSS2940
      DO 181 J=1,21                                                      FLSS2950
      TM$(I,J)=0.D0                                                      FLSS2960
  181 CONTINUE                                                           FLSS2970
                                                                         FLSS2980
END OF INPUT AND VERIFICATION ROUTINE                                    FLSS2990
                                                                         FLSS3000
      DO 300 K=1,NE                                                      FLSS3010
      N1=NODE(K,1)                                                       FLSS3020
      N2=NODE(K,2)                                                       FLSS3030
      N3=NODE(K,3)                                                       FLSS3040
      N4=NODE(K,4)                                                       FLSS3050
      N5=NODE(K,5)                                                       FLSS3060
      N6=NODE(K,6)                                                       FLSS3070
      N7=NODE(K,1)+NN                                                    FLSS3080
      N8=NODE(K,2)+NN                                                    FLSS3090
      N9=NODE(K,3)+NN                                                    FLSS3100
      N10=NODE(K,4)+NN                                                   FLSS3110
      N11=NODE(K,5)+NN                                                   FLSS3120
      N12=NODE(K,6)+NN                                                   FLSS3130
      N13=NCP(NODE(K,1))                                                 FLSS3140
      N14=NCP(NODE(K,3))                                                 FLSS3150
      N15=NCP(NODE(K,5))                                                 FLSS3160
      N16=NODE(K,1)+MM                                                   FLSS3170
      N17=NODE(K,2)+MM                                                   FLSS3180
      N18=NODE(K,3)+MM                                                   FLSS3190
      N19=NODE(K,4)+MM                                                   FLSS3200
      N20=NODE(K,5)+MM                                                   FLSS3210
      N21=NODE(K,6)+MM                                                   FLSS3220
      XC$(1)=XC(NODE(K,1))                                               FLSS3230
      XC$(2)=XC(NODE(K,3))                                               FLSS3240
      XC$(3)=XC(NODE(K,5))                                               FLSS3250
      YC$(1)=YC(NODE(K,1))                                               FLSS3260
      YC$(2)=YC(NODE(K,3))                                               FLSS3270
      YC$(3)=YC(NODE(K,5))                                               FLSS3280
      A=1.D0                                                             FLSS3290
      ZBAR=(YC$(1)+YC$(2)+YC$(3))/3.D0                                   FLSS3300
```

```
      RBAR=(XC$(1)+XC$(2)+XC$(3))/3.DO                                  FLSS3310
      IF(NCASE.EQ.2)A=RBAR                                              FLSS3320
      AA=1.DO                                                           FLSS3330
      IF(NCASE.EQ.2)AA=2.DO*3.14159DO*(RBAR)                            FLSS3340
      A1=XC$(2)*YC$(3)-XC$(3)*YC$(2)                                    FLSS3350
      A2=XC$(3)*YC$(1)-XC$(1)*YC$(3)                                    FLSS3360
      A3=XC$(1)*YC$(2)-XC$(2)*YC$(1)                                    FLSS3370
      B1=YC$(2)-YC$(3)                                                  FLSS3380
      B2=YC$(3)-YC$(1)                                                  FLSS3390
      B3=YC$(1)-YC$(2)                                                  FLSS3400
      C1=XC$(3)-XC$(2)                                                  FLSS3410
      C2=XC$(1)-XC$(3)                                                  FLSS3420
      C3=XC$(2)-XC$(1)                                                  FLSS3430
      DEL=DABS(0.5DO*(XC$(1)*(YC$(2)-YC$(3))+XC$(2)*(YC$(3)-YC$(1))     FLSS3440
     1+XC$(3)*(YC$(1)-YC$(2))))                                         FLSS3450
      CONST=10.90DO*(1.DO*A/(3.DO*DEL))*AA                              FLSS3460
      D1=-B1/6.DO                                                       FLSS3470
      D2=-B2/6.DO                                                       FLSS3480
      D3=-B3/6.DO                                                       FLSS3490
      E1=-C1/6.DO                                                       FLSS3500
      E2=-C2/6.DO                                                       FLSS3510
      E3=-C3/6.DO                                                       FLSS3520
      F1=B1/2520.DO                                                     FLSS3530
      F2=B2/2520.DO                                                     FLSS3540
      F3=B3/2520.DO                                                     FLSS3550
      G1=C1/2520.DO                                                     FLSS3560
      G2=C2/2520.DO                                                     FLSS3570
      G3=C3/2520.DO                                                     FLSS3580
      U1=T1(N1)                                                         FLSS3590
      U2=T1(N2)                                                         FLSS3600
      U3=T1(N3)                                                         FLSS3610
      U4=T1(N4)                                                         FLSS3620
      U5=T1(N5)                                                         FLSS3630
      U6=T1(N6)                                                         FLSS3640
      V1=T1(N7)                                                         FLSS3650
      V2=T1(N8)                                                         FLSS3660
      V3=T1(N9)                                                         FLSS3670
      V4=T1(N10)                                                        FLSS3680
      V5=T1(N11)                                                        FLSS3690
      V6=T1(N12)                                                        FLSS3700
      TM$(1,1)=0.75DO*(B1*B1+C1*C1)*CONST                               FLSS3710
      TM$(1,2)=(B1*B2+C1*C2)*CONST                                      FLSS3720
      TM$(1,3)=-TM$(1,2)*0.25DO                                         FLSS3730
      TM$(1,4)=0.                                                       FLSS3740
      TM$(1,6)=(B1*B3+C1*C3)*CONST                                      FLSS3750
      TM$(1,5)=-TM$(1,6)*.25DO                                          FLSS3760
      TM$(3,3)=.75DO*(B2*B2+C2*C2)*CONST                                FLSS3770
      TM$(3,4)=(B2*B3+C2*C3)*CONST                                      FLSS3780
```

```
      TM$(3,5)=-TM$(3,4)*0.25D0                                         FLSS3790
      TM$(2,1)=TM$(1,2)                                                 FLSS3800
      TM$(2,3)=TM$(1,2)                                                 FLSS3810
      TM$(2,2)=8.D0/3.D0*(TM$(1,1)+TM$(3,3))+2.D0*TM$(1,2)              FLSS3820
      TM$(2,4)=2.D0*TM$(1,6)+TM$(3,4)+TM$(1,2)+4.D0/3.D0*TM$(3,3)       FLSS3830
      TM$(2,5)=0.                                                       FLSS3840
      TM$(2,6)=TM$(1,6)+2.D0*TM$(3,4)+TM$(1,2)+4.D0/3.D0*TM$(1,1)       FLSS3850
      TM$(3,1)=TM$(1,3)                                                 FLSS3860
      TM$(3,2)=TM$(2,3)                                                 FLSS3870
      TM$(3,6)=0.                                                       FLSS3880
      TM$(5,5)=.75D0*(B3*B3+C3*C3)*CONST                                FLSS3890
      TM$(4,1)=TM$(1,4)                                                 FLSS3900
      TM$(4,2)=TM$(2,4)                                                 FLSS3910
      TM$(4,3)=TM$(3,4)                                                 FLSS3920
      TM$(4,4)=8.D0/3.D0*(TM$(3,3)+TM$(5,5))+2.D0*TM$(3,4)              FLSS3930
      TM$(4,5)=TM$(3,4)                                                 FLSS3940
      TM$(4,6)=TM$(1,6)+TM$(3,4)+2.D0*TM$(1,2)+4.D0/3.D0*TM$(5,5)       FLSS3950
      TM$(6,3)=TM$(3,6)                                                 FLSS3960
      TM$(5,1)=TM$(1,5)                                                 FLSS3970
      TM$(5,2)=TM$(2,5)                                                 FLSS3980
      TM$(5,3)=TM$(3,5)                                                 FLSS3990
      TM$(5,4)=TM$(4,5)                                                 FLSS4000
      TM$(5,6)=TM$(1,6)                                                 FLSS4010
      TM$(6,1)=TM$(1,6)                                                 FLSS4020
      TM$(6,2)=TM$(2,6)                                                 FLSS4030
      TM$(6,4)=TM$(4,6)                                                 FLSS4040
      TM$(6,5)=TM$(5,6)                                                 FLSS4050
      TM$(6,6)=8.D0/3.D0*(TM$(5,5)+TM$(1,1))+2.D0*TM$(1,6)              FLSS4060
      IF(NCASE.NE.1) GO TO 3000                                         FLSS4070
                                                                        FLSS4080
C     BEGIN INPUT OF NON-LINEAR TERMS                                   FLSS4090
                                                                        FLSS4100
      TM$(1,1)=TM$(1,1)                                                 FLSS4110
     1-(78.D0*U1+48.D0*U2-9.D0*U3+12.D0*U4-9.D0*U5+48.D0*U6)*F1         FLSS4120
     2              -(78.D0*V1+48.D0*V2-9.D0*V3+12.D0*V4-9.D0*V5+48.D0*V6)FLSS4130
     3*G1                                                               FLSS4140
      TM$(2,1)=TM$(2,1)                                                 FLSS4150
     1-(48.D0*U1+160.D0*U2-32.D0*U3+16.D0*U4-20.D0*U5+80.D0*U6)*F1      FLSS4160
     2              -(48.D0*V1+160.D0*V2-32.D0*V3+16.D0*V4-20.D0*V5+80.D0*FLSS4170
     3V6)*G1                                                            FLSS4180
      TM$(3,1)=TM$(3,1)                                                 FLSS4190
     1-(-9.D0*U1-32.D0*U2-18.D0*U3-16.D0*U4+11.D0*U5-20.D0*U6)*F1       FLSS4200
     2              -(-9.D0*V1-32.D0*V2-18.D0*V3-16.D0*V4+11.D0*V5-20.D0*VFLSS4210
     36)*G1                                                             FLSS4220
      TM$(4,1)=TM$(4,1)                                                 FLSS4230
     1-(12.D0*U1+16.D0*U2-16.D0*U3-96.D0*U4-16.D0*U5+16.D0*U6)*F1       FLSS4240
     2              -(12.D0*V1+16.D0*V2-16.D0*V3-96.D0*V4-16.D0*V5+16.D0*VFLSS4250
     36)*G1                                                             FLSS4260
```

```
      TM$(5,1)=TM$(5,1)                                                 FLSS4270
     1-(-9.D0*U1-20.D0*U2+11.D0*U3-16.D0*U4-18.D0*U5-32.D0*U6)*F1       FLSS4280
     2              -(-9.D0*V1-20.D0*V2+11.D0*V3-16.D0*V4-18.D0*V5-32.D0*VFLSS4290
     36)*G1                                                             FLSS4300
      TM$(6,1)=TM$(6,1)                                                 FLSS4310
     1-(48.D0*U1+80.D0*U2-20.D0*U3+16.D0*U4-32.D0*U5+160.D0*U6)*F1      FLSS4320
     2              -(48.D0*V1+80.D0*V2-20.D0*V3+16.D0*V4-32.D0*V5+160.D0*FLSS4330
     3V6)*G1                                                            FLSS4340
      TM$(1,2)=TM$(1,2)                                                 FLSS4350
     1-(24.D0*U1-32.D0*U2-16.D0*U3-48.D0*U4+4.D0*U5-16.D0*U6)*F1        FLSS4360
     2              -(24.D0*V1-32.D0*V2-16.D0*V3-48.D0*V4+4.D0*V5-16.D0*V6FLSS4370
     3)*G1                       -(120.D0*U1+48.D0*U2-16.D0*U3-16.D0*U4-16.FLSS4380
     4D0*U5+48.D0*U6)*F2                       -(120.D0*V1+48.D0*V2-16.D0*V3-FLSS4390
     516.D0*V4-16.D0*V5+48.D0*V6)*G2                                    FLSS4400
      TM$(2,2)=TM$(2,2)                                                 FLSS4410
     1-(-32.D0*U1+384.D0*U2+48.D0*U3+192.D0*U4-48.D0*U5+128.D0*U6)*F1   FLSS4420
     2              -(-32.D0*V1+384.D0*V2+48.D0*V3+192.D0*V4-48.D0*V5+128.FLSS4430
     3D0*V6)*G1                  -(48.D0*U1+384.D0*U2-32.D0*U3+128.D0*U4-48FLSS4440
     4.D0*U5+192.D0*U6)*F2                     -(48.D0*V1+384.D0*V2-32.D0*V3+FLSS4450
     5128.D0*V4-48.D0*V5+192.D0*V6)*G2                                  FLSS4460
      TM$(3,2)=TM$(3,2)                                                 FLSS4470
     1-(-16.D0*U1+48.D0*U2+120.D0*U3+48.D0*U4-16.D0*U5-16.D0*U6)*F1     FLSS4480
     2              -(-16.D0*V1+48.D0*V2+120.D0*V3+48.D0*V4-16.D0*V5-16.D0FLSS4490
     3*V6)*G1                    -(-16.D0*U1-32.D0*U2+24.D0*U3-16.D0*U4+4.DFLSS4500
     40*U5-48.D0*U6)*F2                        -(-16.D0*V1-32.D0*V2+24.D0*V3-FLSS4510
     516.D0*V4+4.D0*V5-48.D0*V6)*G2                                     FLSS4520
      TM$(4,2)=TM$(4,2)                                                 FLSS4530
     1-(-48.D0*U1+192.D0*U2+48.D0*U3+384.D0*U4-32.D0*U5+128.D0*U6)*F1   FLSS4540
     2              -(-48.D0*V1+192.D0*V2+48.D0*V3+384.D0*V4-32.D0*V5+128.FLSS4550
     3D0*V6)*G1                  -(-16.D0*U1+128.D0*U2-16.D0*U3+128.D0*U4-1FLSS4560
     46.D0*U5+128.D0*U6)*F2                    -(-16.D0*V1+128.D0*V2-16.D0*V3FLSS4570
     5+128.D0*V4-16.D0*V5+128.D0*V6)*G2                                 FLSS4580
      TM$(5,2)=TM$(5,2)                                                 FLSS4590
     1-(4.D0*U1-48.D0*U2-16.D0*U3-32.D0*U4+24.D0*U5-16.D0*U6)*F1        FLSS4600
     2              -(4.D0*V1-48.D0*V2-16.D0*V3-32.D0*V4+24.D0*V5-16.D0*V6FLSS4610
     3)*G1                       -(-16.D0*U1-48.D0*U2+4.D0*U3-16.D0*U4+24.DFLSS4620
     40*U5-32.D0*U6)*F2                        -(-16.D0*V1-48.D0*V2+4.D0*V3-1FLSS4630
     56.D0*V4+24.D0*V5-32.D0*V6)*G2                                     FLSS4640
      TM$(6,2)=TM$(6,2)                                                 FLSS4650
     1-(-16.D0*U1+128.D0*U2-16.D0*U3+128.D0*U4-16.D0*U5+128.D0*U6)*F1   FLSS4660
     2              -(-16.D0*V1+128.D0*V2-16.D0*V3+128.D0*V4-16.D0*V5+128.FLSS4670
     3D0*V6)*G1                  -(48.D0*U1+192.D0*U2-48.D0*U3+128.D0*U4-32FLSS4680
     4.D0*U5+384.D0*U6)*F2                     -(48.D0*V1+192.D0*V2-48.D0*V3+FLSS4690
     5128.D0*V4-32.D0*V5+384.D0*V6)*G2                                  FLSS4700
      TM$(1,3)=TM$(1,3)                                                 FLSS4710
     1-(-18.D0*U1-32.D0*U2-9.D0*U3-20.D0*U4+11.D0*U5-16.D0*U6)*F2       FLSS4720
     2              -(-18.D0*V1-32.D0*V2-9.D0*V3-20.D0*V4+11.D0*V5-16.D0*VFLSS4730
     36)*G2                                                             FLSS4740
```

```
 TM$(2,3)=TM$(2,3)                                                     FLSS4750
1-(-32.D0*U1+160.D0*U2+48.D0*U3+80.D0*U4-20.D0*U5+16.D0*U6)*F2         FLSS4760
2              -(-32.D0*V1+160.D0*V2+48.D0*V3+80.D0*V4-20.D0*V5+16.D0FLSS4770
3*V6)*G2                                                               FLSS4780
 TM$(3,3)=TM$(3,3)                                                     FLSS4790
1-(-9.D0*U1+48.D0*U2+78.D0*U3+48.D0*U4-9.D0*U5+12.D0*U6)*F2            FLSS4800
2              -(-9.D0*V1+48.D0*V2+78.D0*V3+48.D0*V4-9.D0*V5+12.D0*V6FLSS4810
3)*G2                                                                  FLSS4820
 TM$(4,3)=TM$(4,3)                                                     FLSS4830
1-(-20.D0*U1+80.D0*U2+48.D0*U3+160.D0*U4-32.D0*U5+16.D0*U6)*F2         FLSS4840
2              -(-20.D0*V1+80.D0*V2+48.D0*V3+160.D0*V4-32.D0*V5+16.D0FLSS4850
3*V6)*G2                                                               FLSS4860
 TM$(5,3)=TM$(5,3)                                                     FLSS4870
1-(11.D0*U1-20.D0*U2-9.D0*U3-32.D0*U4-18.D0*U5-16.D0*U6)*F2            FLSS4880
2              -(11.D0*V1-20.D0*V2-9.D0*V3-32.D0*V4-18.D0*V5-16.D0*V6FLSS4890
3)*G2                                                                  FLSS4900
 TM$(6,3)=TM$(6,3)                                                     FLSS4910
1-(-16.D0*U1+16.D0*U2+12.D0*U3+16.D0*U4-16.D0*U5-96.D0*U6)*F2          FLSS4920
2              -(-16.D0*V1+16.D0*V2+12.D0*V3+16.D0*V4-16.D0*V5-96.D0*FLSS4930
3V6)*G2                                                                FLSS4940
 TM$(1,4)=TM$(1,4)                                                     FLSS4950
1-(24.D0*U1-16.D0*U2+4.D0*U3-48.D0*U4-16.D0*U5-32.D0*U6)*F2            FLSS4960
2              -(24.D0*V1-16.D0*V2+4.D0*V3-48.D0*V4-16.D0*V5-32.D0*V6FLSS4970
3)*G2                    -(24.D0*U1-32.D0*U2-16.D0*U3-48.D0*U4+4.D0FLSS4980
4*U5-16.D0*U6)*F3                        -(24.D0*V1-32.D0*V2-16.D0*V3-4FLSS4990
58.D0*V4+4.D0*V5-16.D0*V6)*G3                                          FLSS5000
 TM$(2,4)=TM$(2,4)                                                     FLSS5010
1-(-16.D0*U1+128.D0*U2-16.D0*U3+128.D0*U4-16.D0*U5+128.D0*U6)*F2       FLSS5020
2              -(-16.D0*V1+128.D0*V2-16.D0*V3+128.D0*V4-16.D0*V5+128.FLSS5030
3D0*V6)*G2               -(-32.D0*U1+384.D0*U2+48.D0*U3+192.D0*U4-4FLSS5040
48.D0*U5+128.D0*U6)*F3                   -(-32.D0*V1+384.D0*V2+48.D0*V3FLSS5050
5+192.D0*V4-48.D0*V5+128.D0*V6)*G3                                     FLSS5060
 TM$(3,4)=TM$(3,4)                                                     FLSS5070
1-(4.D0*U1-16.D0*U2+24.D0*U3-32.D0*U4+16.D0*U5-48.D0*U6)*F2            FLSS5080
2              -(4.D0*V1-16.D0*V2+24.D0*V3-32.D0*V4+16.D0*V5-48.D0*V6FLSS5090
3)*G2                    -(-16.D0*U1+48.D0*U2+120.D0*U3+48.D0*U4-16FLSS5100
4.D0*U5-16.D0*U6)*F3                     -(-16.D0*V1+48.D0*V2+120.D0*V3FLSS5110
5+48.D0*V4-16.D0*V5-16.D0*V6)*G3                                       FLSS5120
 TM$(4,4)=TM$(4,4)                                                     FLSS5130
1-(-48.D0*U1+128.D0*U2-32.D0*U3+384.D0*U4+48.D0*U5+192.D0*U6)*F2       FLSS5140
2              -(-48.D0*V1+128.D0*V2-32.D0*V3+384.D0*V4+48.D0*V5+192.FLSS5150
3D0*V6)*G2               -(-48.D0*U1+192.D0*U2+48.D0*U3+384.D0*U4-3FLSS5160
42.D0*U5+128.D0*U6)*F3                   -(-48.D0*V1+192.D0*V2+48.D0*V3FLSS5170
5+384.D0*V4-32.D0*V5+128.D0*V6)*G3                                     FLSS5180
 TM$(5,4)=TM$(5,4)                                                     FLSS5190
1-(-16.D0*U1-16.D0*U2-16.D0*U3+48.D0*U4+120.D0*U5+48.D0*U6)*F2         FLSS5200
2              -(-16.D0*V1-16.D0*V2-16.D0*V3+48.D0*V4+120.D0*V5+48.D0FLSS5210
3*V6)*G2                 -(4.D0*U1-48.D0*U2-16.D0*U3-32.D0*U4+24.D0FLSS5220
```

```
4*U5-16.D0*U6)*F3                              -(4.D0*V1-48.D0*V2-16.D0*V3-32FLSS5230
5.D0*V4+24.D0*V5-16.D0*V6)*G3                                                 FLSS5240
 TM$(6,4)=TM$(6,4)                                                            FLSS5250
1-(-32.D0*U1+128.D0*U2-48.D0*U3+192.D0*U4+48.D0*U5+384.D0*U6)*F2              FLSS5260
2          -(-32.D0*V1+128.D0*V2-48.D0*V3+192.D0*V4+48.D0*V5+384.FLSS5270
3D0*V6)*G2                -(-16.D0*U1+128.D0*U2-16.D0*U3+128.D0*U4-1FLSS5280
46.D0*U5+128.D0*U6)*F3                         -(-16.D0*V1+128.D0*V2-16.D0*V3FLSS5290
5+128.D0*V4-16.D0*V5+128.D0*V6)*G3                                            FLSS5300
 TM$(1,5)=TM$(1,5)                                                            FLSS5310
1-(-18.D0*U1-16.D0*U2+11.D0*U3-20.D0*U4-9.D0*U5-32.D0*U6)*F3                  FLSS5320
2          -(-18.D0*V1-16.D0*V2+11.D0*V3-20.D0*V4-9.D0*V5-32.D0*VFLSS5330
36)*G3                                                                        FLSS5340
 TM$(2,5)=TM$(2,5)                                                            FLSS5350
1-(-16.D0*U1-96.D0*U2-16.D0*U3+16.D0*U4+12.D0*U5+16.D0*U6)*F3                 FLSS5360
2          -(-16.D0*V1-96.D0*V2-16.D0*V3+16.D0*V4+12.D0*V5+16.D0*FLSS5370
3V6)*G3                                                                       FLSS5380
 TM$(3,5)=TM$(3,5)                                                            FLSS5390
1-(11.D0*U1-16.D0*U2-18.D0*U3-32.D0*U4-9.D0*U5-20.D0*U6)*F3                   FLSS5400
2          -(11.D0*V1-16.D0*V2-18.D0*V3-32.D0*V4-9.D0*V5-20.D0*V6FLSS5410
3)*G3                                                                         FLSS5420
 TM$(4,5)=TM$(4,5)                                                            FLSS5430
1-(-20.D0*U1+16.D0*U2-32.D0*U3+160.D0*U4+48.D0*U5+80.D0*U6)*F3                FLSS5440
2          -(-20.D0*V1+16.D0*V2-32.D0*V3+160.D0*V4+48.D0*V5+80.D0FLSS5450
3*V6)*G3                                                                      FLSS5460
 TM$(5,5)=TM$(5,5)                                                            FLSS5470
1-(-9.D0*U1+12.D0*U2-9.D0*U3+48.D0*U4+78.D0*U5+48.D0*U6)*F3                   FLSS5480
2          -(-9.D0*V1+12.D0*V2-9.D0*V3+48.D0*V4+78.D0*V5+48.D0*V6FLSS5490
3)*G3                                                                         FLSS5500
 TM$(6,5)=TM$(6,5)                                                            FLSS5510
1-(-32.D0*U1+16.D0*U2-20.D0*U3+80.D0*U4+48.D0*U5+160.D0*U6)*F3                FLSS5520
2          -(-32.D0*V1+16.D0*V2-20.D0*V3+80.D0*V4+48.D0*V5+160.D0FLSS5530
3*V6)*G3                                                                      FLSS5540
 TM$(1,6)=TM$(1,6)                                                            FLSS5550
1-(24.D0*U1-16.D0*U2+4.D0*U3-48.D0*U4-16.D0*U5-32.D0*U6)*F1                   FLSS5560
2          -(24.D0*V1-16.D0*V2+4.D0*V3-48.D0*V4-16.D0*V5-32.D0*V6FLSS5570
3)*G1                     -(120.D0*U1+48.D0*U2-16.D0*U3-16.D0*U4-16.FLSS5580
4D0*U5+48.D0*U6)*F3                            -(120.D0*V1+48.D0*V2-16.D0*V3-FLSS5590
516.D0*V4-16.D0*V5+48.D0*V6)*G3                                               FLSS5600
 TM$(2,6)=TM$(2,6)                                                            FLSS5610
1-(-16.D0*U1+128.D0*U2-16.D0*U3+128.D0*U4-16.D0*U5+128.D0*U6)*F1              FLSS5620
2          -(-16.D0*V1+128.D0*V2-16.D0*V3+128.D0*V4-16.D0*V5+128.FLSS5630
3D0*V6)*G1                -(48.D0*U1+384.D0*U2-32.D0*U3+128.D0*U4-48FLSS5640
4.D0*U5+192.D0*U6)*F3                          -(48.D0*V1+384.D0*V2-32.D0*V3+FLSS5650
5128.D0*V4-48.D0*V5+192.D0*V6)*G3                                             FLSS5660
 TM$(3,6)=TM$(3,6)                                                            FLSS5670
1-(4.D0*U1-16.D0*U2+24.D0*U3-32.D0*U4-16.D0*U5-48.D0*U6)*F1                   FLSS5680
2          -(4.D0*V1-16.D0*V2+24.D0*V3-32.D0*V4-16.D0*V5-48.D0*V6FLSS5690
3)*G1                     -(-16.D0*U1-32.D0*U2+24.D0*U3-16.D0*U4+4.DFLSS5700
```

```
     40*U5-48.D0*U6)*F3                        -(-16.D0*V1-32.D0*V2+24.D0*V3-FLSS5710
     516.D0*V4+4.D0*V5-48.D0*V6)*G3                                          FLSS5720
      TM$(4,6)=TM$(4,6)                                                      FLSS5730
     1-(-48.D0*U1+128.D0*U2-32.D0*U3+384.D0*U4+48.D0*U5+192.D0*U6)*F1        FLSS5740
     2          -(-48.D0*V1+128.D0*V2-32.D0*V3+384.D0*V4+48.D0*V5+192.FLSS5750
     3D0*V6)*G1              -(-16.D0*U1+128.D0*U2-16.D0*U3+128.D0*U4-1FLSS5760
     46.D0*U5+128.D0*U6)*F3                    -(-16.D0*V1+128.D0*V2-16.D0*V3FLSS5770
     5+128.D0*V4-16.D0*V5+128.D0*V6)*G3                                      FLSS5780
      TM$(5,6)=TM$(5,6)                                                      FLSS5790
     1-(-16.D0*U1-16.D0*U2-16.D0*U3+48.D0*U4+120.D0*U5+48.D0*U6)*F1          FLSS5800
     2          -(-16.D0*V1-16.D0*V2-16.D0*V3+48.D0*V4+120.D0*V5+48.D0FLSS5810
     3*V6)*G1                -(-16.D0*U1-48.D0*U2+4.D0*U3-16.D0*U4+24.DFLSS5820
     40*U5-32.D0*U6)*F3                        -(-16.D0*V1-48.D0*V2+4.D0*V3-1FLSS5830
     56.D0*V4+24.D0*V5-32.D0*V6)*G3                                          FLSS5840
      TM$(6,6)=TM$(6,6)                                                      FLSS5850
     1-(-32.D0*U1+128.D0*U2-48.D0*U3+192.D0*U4+48.D0*U5+384.D0*U6)*F1        FLSS5860
     2          -(-32.D0*V1+128.D0*V2-48.D0*V3+192.D0*V4+48.D0*V5+384.FLSS5870
     3D0*V6)*G1              -(48.D0*U1+192.D0*U2-48.D0*U3+128.D0*U4-32FLSS5880
     4.D0*U5+384.D0*U6)*F3                     -(48.D0*V1+192.D0*V2-48.D0*V3+FLSS5890
     5128.D0*V4-32.D0*V5+384.D0*V6)*G3                                       FLSS5900
                                                                             FLSS5910
C THIS ENCS ADDITION OF NON-LINEAR TERMS TO THE LOCAL ARRAY                   FLSS5920
                                                                             FLSS5930
 3000 CONTINUE                                                               FLSS5940
      TM$(7,7)=TM$(1,1)                                                      FLSS5950
      TM$(7,8)=TM$(1,2)                                                      FLSS5960
      TM$(7,9)=TM$(1,3)                                                      FLSS5970
      TM$(7,10)=TM$(1,4)                                                     FLSS5980
      TM$(7,11)=TM$(1,5)                                                     FLSS5990
      TM$(7,12)=TM$(1,6)                                                     FLSS6000
      TM$(8,7)=TM$(2,1)                                                      FLSS6010
      TM$(8,8)=TM$(2,2)                                                      FLSS6020
      TM$(8,9)=TM$(2,3)                                                      FLSS6030
      TM$(8,10)=TM$(2,4)                                                     FLSS6040
      TM$(8,11)=TM$(2,5)                                                     FLSS6050
      TM$(8,12)=TM$(2,6)                                                     FLSS6060
      TM$(9,7)=TM$(3,1)                                                      FLSS6070
      TM$(9,8)=TM$(3,2)                                                      FLSS6080
      TM$(9,9)=TM$(3,3)                                                      FLSS6090
      TM$(9,10)=TM$(3,4)                                                     FLSS6100
      TM$(9,11)=TM$(3,5)                                                     FLSS6110
      TM$(9,12)=TM$(3,6)                                                     FLSS6120
      TM$(10,7)=TM$(4,1)                                                     FLSS6130
      TM$(10,8)=TM$(4,2)                                                     FLSS6140
      TM$(10,9)=TM$(4,3)                                                     FLSS6150
      TM$(10,10)=TM$(4,4)                                                    FLSS6160
      TM$(10,11)=TM$(4,5)                                                    FLSS6170
      TM$(10,12)=TM$(4,6)                                                    FLSS6180
```

```
      TM$(11,7)=TM$(5,1)                                               FLSS6190
      TM$(11,8)=TM$(5,2)                                               FLSS6200
      TM$(11,9)=TM$(5,3)                                               FLSS6210
      TM$(11,10)=TM$(5,4)                                              FLSS6220
      TM$(11,11)=TM$(5,5)                                              FLSS6230
      TM$(11,12)=TM$(5,6)                                              FLSS6240
      TM$(12,7)=TM$(6,1)                                               FLSS6250
      TM$(12,8)=TM$(6,2)                                               FLSS6260
      TM$(12,9)=TM$(6,3)                                               FLSS6270
      TM$(12,10)=TM$(6,4)                                              FLSS6280
      TM$(12,11)=TM$(6,5)                                              FLSS6290
      TM$(12,12)=TM$(6,6)                                              FLSS6300
      TM$(13,1)=D1                                                     FLSS6310
      TM$(14,1)=0.D0                                                   FLSS6320
      TM$(15,1)=0.D0                                                   FLSS6330
      TM$(13,2)=D1+2.D0*D2                                             FLSS6340
      TM$(14,2)=2.D0*D1+D2                                             FLSS6350
      TM$(15,2)=D1+D2                                                  FLSS6360
      TM$(13,3)=0.D0                                                   FLSS6370
      TM$(14,3)=D2                                                     FLSS6380
      TM$(15,3)=0.D0                                                   FLSS6390
      TM$(13,4)=D2+D3                                                  FLSS6400
      TM$(14,4)=D2+2.D0*D3                                             FLSS6410
      TM$(15,4)=2.D0*D2+D3                                             FLSS6420
      TM$(13,5)=0.D0                                                   FLSS6430
      TM$(14,5)=0.D0                                                   FLSS6440
      TM$(15,5)=D3                                                     FLSS6450
      TM$(13,6)=D1+2.D0*D3                                             FLSS6460
      TM$(14,6)=D1+D3                                                  FLSS6470
      TM$(15,6)=2.D0*D1+D3                                             FLSS6480
      TM$(13,7)=E1                                                     FLSS6490
      TM$(14,7)=0.D0                                                   FLSS6500
      TM$(15,7)=0.D0                                                   FLSS6510
      TM$(13,8)=E1+2.D0*E2                                             FLSS6520
      TM$(14,8)=2.D0*E1+E2                                             FLSS6530
      TM$(15,8)=E1+E2                                                  FLSS6540
      TM$(13,9)=0.D0                                                   FLSS6550
      TM$(14,9)=E2                                                     FLSS6560
      TM$(15,9)=0.0                                                    FLSS6570
      TM$(13,10)=E2+E3                                                 FLSS6580
      TM$(14,10)=E2+2.D0*E3                                            FLSS6590
      TM$(15,10)=E2+E2+E3                                              FLSS6600
      TM$(13,11)=0.D0                                                  FLSS6610
      TM$(14,11)=0.D0                                                  FLSS6620
      TM$(15,11)=E3                                                    FLSS6630
      TM$(13,12)=E1+E3+E3                                              FLSS6640
      TM$(14,12)=E1+E3                                                 FLSS6650
      TM$(15,12)=2.D0*E1+E3                                            FLSS6660
```

```
      CONST4=0.94375D0                                              FLSS6670
      TM$(1,13)=D1*CONST4                                           FLSS6680
      TM$(1,14)=0.D0                                                FLSS6690
      TM$(1,15)=0.D0                                                FLSS6700
      TM$(2,13)=(D1+2.D0*D2)*CONST4                                 FLSS6710
      TM$(2,14)=(2.D0*D1+D2)*CONST4                                 FLSS6720
      TM$(2,15)=(D1+D2)*CONST4                                      FLSS6730
      TM$(3,13)=0.D0                                                FLSS6740
      TM$(3,14)=D2*CONST4                                           FLSS6750
      TM$(3,15)=0.D0                                                FLSS6760
      TM$(4,13)=(D2+D3)*CONST4                                      FLSS6770
      TM$(4,14)=(D2+2.D0*D3)*CONST4                                 FLSS6780
      TM$(4,15)=(2.D0*D2+D3)*CONST4                                 FLSS6790
      TM$(5,13)=0.D0                                                FLSS6800
      TM$(5,14)=0.D0                                                FLSS6810
      TM$(5,15)=D3*CONST4                                           FLSS6820
      TM$(6,13)=(D1+2.D0*D3)*CONST4                                 FLSS6830
      TM$(6,14)=(D1+D3)*CONST4                                      FLSS6840
      TM$(6,15)=(2.D0*D1+D3)*CONST4                                 FLSS6850
      TM$(7,13)=E1*CONST4                                           FLSS6860
      TM$(7,14)=0.D0                                                FLSS6870
      TM$(7,15)=0.D0                                                FLSS6880
      TM$(8,13)=(E1+2.D0*E2)*CONST4                                 FLSS6890
      TM$(8,14)=(2.D0*E1+E2)*CONST4                                 FLSS6900
      TM$(8,15)=(E1+E2)*CONST4                                      FLSS6910
      TM$(9,13)=0.D0                                                FLSS6920
      TM$(9,14)=E2*CONST4                                           FLSS6930
      TM$(9,15)=0.D0                                                FLSS6940
      TM$(10,13)=(E2+E3)*CONST4                                     FLSS6950
      TM$(10,14)=(E2+2.D0*E3)*CONST4                                FLSS6960
      TM$(10,15)=(E2+E2+E3)*CONST4                                  FLSS6970
      TM$(11,13)=0.D0                                               FLSS6980
      TM$(11,14)=0.D0                                               FLSS6990
      TM$(11,15)=E3*CONST4                                          FLSS7000
      TM$(12,13)=(E1+2.D0*E3)*CONST4                                FLSS7010
      TM$(12,14)=(E1+E3)*CONST4                                     FLSS7020
      TM$(12,15)=(2.D0*E1+E3)*CONST4                                FLSS7030
      CONST2=3.2981D0                                               FLSS7040
      TM$(1,16)=1.D0*CONST2                                         FLSS7050
      TM$(2,17)=1.D0*CONST2                                         FLSS7060
      TM$(3,18)=1.D0*CONST2                                         FLSS7070
      TM$(4,19)=1.D0*CONST2                                         FLSS7080
      TM$(5,20)=1.D0*CONST2                                         FLSS7090
      TM$(6,21)=1.D0*CONST2                                         FLSS7100
                                                                    FLSS7110
C     ALPHA1=ALPHA/VISCOSITY                                        FLSS7120
                                                                    FLSS7130
      ALPHA1=9.4454D-05                                             FLSS7140
```

```
      CONST1=ALPHA1                                                     FLSS7150
      TM$(16,16)=TM$(1,1)*CONST1                                        FLSS7160
      TM$(16,17)=TM$(1,2)*CONST1                                        FLSS7170
      TM$(16,18)=TM$(1,3)*CONST1                                        FLSS7180
      TM$(16,19)=TM$(1,4)*CONST1                                        FLSS7190
      TM$(16,20)=TM$(1,5)*CONST1                                        FLSS7200
      TM$(16,21)=TM$(1,6)*CONST1                                        FLSS7210
      TM$(17,16)=TM$(2,1)*CONST1                                        FLSS7220
      TM$(17,17)=TM$(2,2)*CONST1                                        FLSS7230
      TM$(17,18)=TM$(2,3)*CONST1                                        FLSS7240
      TM$(17,19)=TM$(2,4)*CONST1                                        FLSS7250
      TM$(17,20)=TM$(2,5)*CONST1                                        FLSS7260
      TM$(17,21)=TM$(2,6)*CONST1                                        FLSS7270
      TM$(18,16)=TM$(3,1)*CONST1                                        FLSS7280
      TM$(18,17)=TM$(3,2)*CONST1                                        FLSS7290
      TM$(18,18)=TM$(3,3)*CONST1                                        FLSS7300
      TM$(18,19)=TM$(3,4)*CONST1                                        FLSS7310
      TM$(18,20)=TM$(3,5)*CONST1                                        FLSS7320
      TM$(18,21)=TM$(3,6)*CONST1                                        FLSS7330
      TM$(19,16)=TM$(4,1)*CONST1                                        FLSS7340
      TM$(19,17)=TM$(4,2)*CONST1                                        FLSS7350
      TM$(19,18)=TM$(4,3)*CONST1                                        FLSS7360
      TM$(19,19)=TM$(4,4)*CONST1                                        FLSS7370
      TM$(19,20)=TM$(4,5)*CONST1                                        FLSS7380
      TM$(19,21)=TM$(4,6)*CONST1                                        FLSS7390
      TM$(20,16)=TM$(5,1)*CONST1                                        FLSS7400
      TM$(20,17)=TM$(5,2)*CONST1                                        FLSS7410
      TM$(20,18)=TM$(5,3)*CONST1                                        FLSS7420
      TM$(20,19)=TM$(5,4)*CONST1                                        FLSS7430
      TM$(20,20)=TM$(5,5)*CONST1                                        FLSS7440
      TM$(20,21)=TM$(5,6)*CONST1                                        FLSS7450
      TM$(21,16)=TM$(6,1)*CONST1                                        FLSS7460
      TM$(21,17)=TM$(6,2)*CONST1                                        FLSS7470
      TM$(21,18)=TM$(6,3)*CONST1                                        FLSS7480
      TM$(21,19)=TM$(6,4)*CONST1                                        FLSS7490
      TM$(21,20)=TM$(6,5)*CONST1                                        FLSS7500
      TM$(21,21)=TM$(6,6)*CONST1                                        FLSS7510
  185 CONTINUE                                                          FLSS7520
      N(1)=N1                                                           FLSS7530
      N(2)=N2                                                           FLSS7540
      N(3)=N3                                                           FLSS7550
      N(4)=N4                                                           FLSS7560
      N(5)=N5                                                           FLSS7570
      N(6)=N6                                                           FLSS7580
      N(7)=N7                                                           FLSS7590
      N(8)=N8                                                           FLSS7600
      N(9)=N9                                                           FLSS7610
      N(10)=N10                                                         FLSS7620
```

```
      N(11)=N11                                                         FLSS7630
      N(12)=N12                                                         FLSS7640
      N(13)=N13                                                         FLSS7650
      N(14)=N14                                                         FLSS7660
      N(15)=N15                                                         FLSS7670
      N(16)=N16                                                         FLSS7680
      N(17)=N17                                                         FLSS7690
      N(18)=N18                                                         FLSS7700
      N(19)=N19                                                         FLSS7710
      N(20)=N20                                                         FLSS7720
      N(21)=N21                                                         FLSS7730
      DO 200 I$=1,21                                                    FLSS7740
      I=N(I$)                                                           FLSS7750
      DO 200 J$=1,21                                                    FLSS7760
      J=N(J$)                                                           FLSS7770
      TM(I,J)=TM(I,J)+TM$(I$,J$)                                        FLSS7780
  200 CONTINUE                                                          FLSS7790
  300 CONTINUE                                                          FLSS7800
      IF(NNQXY.EQ.0) GO TO 310                                          FLSS7810
      DO 310 I=1,NNQXY                                                  FLSS7820
      RHS(NQS(I))=RHS(NQS(I))+Q(NQS(I))+65.962D0                        FLSS7830
      RHS(NQS(I)+NN)=RHS(NQS(I)+NN)+Q(NQS(I)+NN)                        FLSS7840
  310 CONTINUE                                                          FLSS7850
      IF(NNQZC.EQ.0) GO TO 312                                          FLSS7860
      DO 312 I=1,NNQZC                                                  FLSS7870
      RHS(NQS(NNQXY+I)+2*NN)=RHS(NQS(NNQXY+I)+2*NN)+Q(NQS(NNQXY+I)+2*NN)FLSS7880
  312 CONTINUE                                                          FLSS7890
      IF(NNQZ.EQ.0) GO TO 311                                           FLSS7900
      DO 311 I=1,NNQZ                                                   FLSS7910
      RHS(NQS(NNQXY+NNQZC+I)+MM)=RHS(NQS(NNQXY+NNQZC+I)+MM)+            FLSS7920
     1Q(NQS(NNQXY+NNQZC+I)+MM)                                          FLSS7930
  311 CONTINUE                                                          FLSS7940
                                                                        FLSS7950
    MODIFICATION OF RHS FOR TM BOUNDARY CONDITIONS                      FLSS7960
                                                                        FLSS7970
      DO 315 I=1,MMM                                                    FLSS7980
      DO 315 J=1,NTOTVP                                                 FLSS7990
      JX=NVIS(J)                                                        FLSS8000
      RHS(I)=RHS(I)-TM(I,JX)*X(JX)                                      FLSS8010
      TM(I,JX)=0.D0                                                     FLSS8020
      TM(JX,I)=0.D0                                                     FLSS8030
      TM(JX,JX)=1.D0                                                    FLSS8040
      RHS(JX)=X(JX)                                                     FLSS8050
  315 CONTINUE                                                          FLSS8060
      M=1                                                               FLSS8070
      ND=117                                                            FLSS8080
      IA=117                                                            FLSS8090
      IDGT=0                                                            FLSS8100
```

```
      CALL LEQT2F (TM,M,ND,IA,RHS,IDGT,WKAREA,IER)                      FLSS8110
      WRITE(NWRITE,2000)                                                FLSS8120
      DO 322  J=1,MMM                                                   FLSS8130
      TDIFF=DABS(T1(J)-X(J))                                            FLSS8140
      EPSLN=1.D-06                                                      FLSS8150
      IF(TDIFF-EPSLN) 322,324,324                                       FLSS8160
  322 CONTINUE                                                          FLSS8170
  324 CONTINUE                                                          FLSS8180
      DO 360  I=1,MMM                                                   FLSS8190
      X(I)=RHS(I)                                                       FLSS8200
      T1(I)=X(I)                                                        FLSS8210
      WRITE (NWRITE,2005) I,X(I)                                        FLSS8220
  360 CONTINUE                                                          FLSS8230
      IF(J.NE.MMM) GO TO 177                                            FLSS8240
      WRITE(NWRITE,2021)                                                FLSS8250
      WRITE(NWRITE,2025)                                                FLSS8260
      WRITE(NWRITE,2030)                                                FLSS8270
      WRITE(NWRITE,2035)                                                FLSS8280
      WRITE(NWRITE,2040)                                                FLSS8290
      WRITE(NWRITE,2045)                                                FLSS8300
      WRITE(NWRITE,2050)                                                FLSS8310
  500 FORMAT(I10)                                                       FLSS8330
  600 FORMAT(1H1,18X,'STEADY STATE FLUID MECHANICS PROBLEM',////)
 1005 FORMAT(3I10)                                                      FLSS8340
 1006 FORMAT(6X,A4,I10,2F10.0)                                          FLSS8350
 1010 FORMAT(7I10)                                                      FLSS8360
 1015 FORMAT(6X,A4,I10,F10.0)                                           FLSS8370
 1016 FORMAT(6X,A4,I10)                                                 FLSS8380
 1020 FORMAT(I10,2F10.0)                                                FLSS8390
 1025 FORMAT(6X,A4,I10,F10.0)                                           FLSS8400
 1030 FORMAT(6X,A4,2I10,2F10.0)                                         FLSS8410
 1034 FORMAT(5X,'NNQZC=',I3,//)                                         FLSS8420
 1035 FORMAT(5X,'NO. OF NODES=',I3,//,5X,'NO. OF ELEMENTS=',I3,//,      FLSS8430
     15X,'NO. OF CORNER NODES=',I3,///)                                 FLSS8440
 1036 FORMAT(5X,'NNVELS=',I3,//)                                        FLSS8450
 1037 FORMAT(5X,'NNQXY=',I3,//)                                         FLSS8460
 1038 FORMAT(5X,'NNPS=',I3,//)                                          FLSS8470
 1039 FORMAT(5X,'NNTS=',I3,//)                                          FLSS8480
 1040 FORMAT(5X,'NNQZ=',I3,///)                                         FLSS8490
 1041 FORMAT(5X,'SUMMARY OF NODAL COORDINATES',//,                      FLSS8500
     17X,'I',12X'X(I)',13X,'Y(I)',//)                                   FLSS8510
 1045 FORMAT(5X,I3,2(7X,F10.3))                                         FLSS8520
 1050 FORMAT(//,5X,'LISTING OF SYSTEM TOPOLOGY',//,5X,
     1'ELEMENT NUMBER',20X,'NODE NUMBERS',//)                           FLSS8540
 1055 FORMAT(5X,I3,10X,6(5X,I3))                                        FLSS8550
 1060 FORMAT(//,7X,'NODES WHERE VELOCITIES ARE SPECIFIED',              FLSS8560
     1//,8X,'I',5X,'NODE',5X,'U VELOCITY',5X,'V VELOCITY',//)           FLSS8570
 1065 FORMAT(2X,2(4X,I3),3X,F12.3,3X,F12.3)                             FLSS8580
```

```
1070 FORMAT(//,5X,'NODES WHERE QX AND QY ARE SPECIFIED',                FLSS8590
    1       //,2X,'      I    NODE',1X,'        QX',10X,'   QY',//)      FLSS8600
1075 FORMAT(5X,I3,2(10X,F12.3))                                          FLSS8610
1080 FORMAT(//,5X,'NODES WHERE PRESSURE IS SPECIFIED',                   FLSS8620
    1       //,9X,'I',4X,'NODE',13X,'PRESSURE',//)                        FLSS8630
1081 FORMAT(//,5X,'NODES WHERE TEMPERATURE IS SPECIFIED',                FLSS8640
    1       //,9X,'I',4X,'NODE',12X,'TEMPERATURE',//)                     FLSS8650
1082 FORMAT(//,5X,'NODES WHERE HEAT FLUX QZ IS SPECIFIED',               FLSS8660
    1       //,9X,'I',4X,'NODE',12X,'HEAT FLUX',//)                       FLSS8670
1083 FORMAT(//,5X,'NODES WHERE QZC IS SPECIFIED',                        FLSS8680
    1       //,9X,'I',4X,'NODE',15X,'QZC',//)                             FLSS8690
1085 FORMAT(7X,I3,3X,I3,10X,F12.3)                                       FLSS8700
1090 FORMAT(////,5X,'NODAL VARIABLE IS THE U-VELOCITY AT NODES 1 - 35;'
    1,//,5X,'THE V-VELOCITY AT NODES 36 - 70;',
    2//,5X,'THE PRESSURE AT NODES 71 - 82;',
    3//,5X,'AND THE TEMPERATURE AT NODES 83 - 117.',////)
1091 FORMAT(////,5X,'THE FIRST SEQUENCE OF 117 NODAL VARIABLES',
    1//,5X,'REPRESENTS A LINEAR,STEADY STATE SYSTEM;',
    2//,5X,'WHILE THE SECOND SET OF THE 117 VALUES CORRESPONDS',
    3//,5X,'TO A NONLINEAR ANALYSIS OF THE SOLUTION DOMAIN.',////)
1095 FORMAT(5X,3(3X,I3),2(5X,F12.3))                                     FLSS8710
2000 FORMAT(///,5X,'NODE NO.',6X,'NODE VARIABLE',//)                     FLSS8720
2005 FORMAT(9X,I3,5X,D17.10,/)                                           FLSS8730
2015 FORMAT(1H1,21X,'THIS IS A 3-D AXISYMMETRIC PROBLEM',//)             FLSS8740
2020 FORMAT(21X,'THIS IS A 2-D NONLINEAR PROBLEM',//)                    FLSS8750
2021 FORMAT(/////////)                                                   FLSS8760
2025 FORMAT(///,10X,'THE KINEMATIC VISCOSITY OF FLUID 50-HB-3520 AT 20 FLSS8770
    1DEGREES C. =',//,
    221X,'10.90 SQ.CM/SEC',//)
2030 FORMAT(///,10X,'THE DENSITY OF 50-HB-3520 AT 20 DEGREES C. = 1.059FLSS8790
    16 GM/CC',//)                                                         FLSS8800
2035 FORMAT(///,10X,'THE COEFF. OF THERMAL EXPANSION OF 50-HB-3520 AT 2FLSS8810
    10 DEGREES C. =',//,
    221X,'0.002278/DEGREE C.',//)
2040 FORMAT(///,10X,'THE THERMAL DIFFUSIVITY OF 50-HB-3520 AT 20 DEGREEFLSS8830
    1S C. =',//,
    221X,'0.00103 SQ.CM/SEC',//)
2045 FORMAT(///,10X,'THE GRASHOF NUMBER (GR(L)) =(G*B*L**3*(TH-TC))/V**FLSS8850
    12 = 946.4',//)                                                       FLSS8860
2050 FORMAT(///,10X,'THE U VELOCITY FORCING FUNCTION,G*B*T(INITIAL), = FLSS8870
    165.962 CM/SQ.SEC',//,1H1)                                            FLSS8880
2055 FORMAT(////,5X,'THE SPECIFIED WALL PRESSURES',
    1//,5X,'ARE NORMALIZED TO ONE (1) ATMOSPHERE,',
    2//,5X,'THAT IS, 1014000 DYNES/SQ.CM',
    3//,5X,'(ALL PARAMETER VALUES ARE IN CGS UNITS).',////)
     STOP                                                                 FLSS8910
     END                                                                  FLSS8920
```

TIME-DEPENDENT FLUID MECHANICS PROBLEM

```
      IMPLICIT REAL*8(A-H,O-Z,$)
      DATA NREAD/5/

THE U VELOCITY IS IN THE FIRST NN POSITIONS OF X(I)
THE V VELOCITY IS IN THE SECOND NN POSITIONS OF X I.E. X(I+NN)
THE P PRESSURE IS IN THE NN+NN+I POSITIONS OF X I.E. X(I+NN+NN)
THE T TEMPERATURE IS IN THE NN+NN+NNCN+I PPOSITIONS OF X I.E.
X(I+NN+NN+NNCN)
THERE ARE NNCN PRESSURE NODES (NNCN=NUMBER OF CORNER NODES)
TM MUST BE DIMENSIONED 3*NN+NNCN X 3*NN+NNCN


      DATA NWRITE/6/
      DATA STOP/'STOP'/
      DIMENSION XC(125),YC(125),NODE(125,6),NVS(125),NCN(125)
      DIMENSION X(117),NVIS(117),NCP(125),NPS(125),Q(117)
      DIMENSION NQS(125),T1(117)
      DIMENSION TM$(21,21),N(21),NQIS(117)
      DIMENSION RP$(6),ZP$(6)
      DIMENSION XC$(3),YC$(3)
      DIMENSION CD$(21,21)
      DIMENSION Y(7,117),W(117,132)
      COMMON CD(117,117),TM(117,117),RHS(117)

SPECIFY WHETHER TWO DIMENSIONAL, INCLUDING NON-LINEAR TERMS,
(NCASE=1) OR AXISYMMETRIC (NCASE=2)

      READ(NREAD,500)NCASE
      IF(NCASE.EQ.1)GO TO 5
      WRITE(NWRITE,2015)
      GO TO 6
    5 WRITE(NWRITE,2020)
    6 CONTINUE

THE FIRST PART OF THE PROGRAM CAN BE CONSIDERED AS AN INPUT ROUTINE
```

```
IN WHICH LINES 049 TO 0303 ARE INPUT VERIFICATION OF ALL DATA.     TIME0038
SUCH A SECTION WOULD BE PART OF ANY FINITE ELEMENT PROGRAM.        TIME0040
                                                                   TIME0041
                                                                   TIME0042
READ IN NUMBER OF NODES AND ELEMENTS AND NO. OF CORNER NODES       TIME0043
                                                                   TIME0044
      READ(NREAD,1005)NN,NE,NNCN                                   TIME0045
                                                                   TIME0046
INITIALIZE ALL PARAMETERS                                          TIME0047
                                                                   TIME0048
      MM=2*NN+NNCN                                                 TIME0049
      MMM=3*NN+NNCN                                                TIME0050
      DO 50 I=1,MMM                                                TIME0051
      XC(I)=0.D0                                                   TIME0052
      YC(I)=0.D0                                                   TIME0053
      NVS(I)=0                                                     TIME0054
      NCN(I)=0                                                     TIME0055
      NCP(I)=0                                                     TIME0056
      NPS(I)=0                                                     TIME0057
      NQS(I)=0                                                     TIME0058
      DO 50 J=1,6                                                  TIME0059
      NODE(I,J)=0                                                  TIME0060
   50 CONTINUE                                                     TIME0061
      DO 51 I=1,MMM                                                TIME0062
      T1(I)=0.D0                                                   TIME0063
      Y(1,I)=0.D0                                                  TIME0064
      X(I)=0.D0                                                    TIME0065
      NVIS(I)=0                                                    TIME0066
      Q(I)=0.D0                                                    TIME0067
      NQIS(I)=0                                                    TIME0068
      RHS(I)=0.D0                                                  TIME0069
      DO 51 J=1,MMM                                                TIME0070
      TM(I,J)=0.D0                                                 TIME0071
      CC(I,J)=0.D0                                                 TIME0072
   51 CONTINUE                                                     TIME0073
      DO 52 I=1,21                                                 TIME0074
      N(I)=0                                                       TIME0075
      DO 52 J=1,21                                                 TIME0076
      TM$(I,J)=0.D0                                                TIME0077
      CD$(I,J)=0.D0                                                TIME0078
   52 CONTINUE                                                     TIME0079
      DO 53 I=1,6                                                  TIME0080
      RP$(I)=0.D0                                                  TIME0081
      ZP$(I)=0.D0                                                  TIME0082
   53 CONTINUE                                                     TIME0083
      DO 54 I=1,3                                                  TIME0084
      XC$(I)=0.D0                                                  TIME0085
      YC$(I)=0.D0                                                  TIME0086
```

```
   54 CONTINUE

 READ NODE NUMBER AND COORDINATES

      DO 100 J=1,NN
      READ(NREAD,1006)WORD,I,XC(I),YC(I)
      IF(WORD.EQ.STOP) GO TO 101
      NCN(J)=I
  100 CONTINUE
  101 NNCN=J-1

 THE ARRAY NCP(J) GENERATES THE GLOBAL PRESSURE INDICES (P1,P2.ETC.)
 THUS PRESSURE NODES ARE LABELED AS CORNER NODES AND ARE INPUTED
 WHEN CNE INPUTS A GLOBAL CORNER NODE FOR J

      DO 107 J=1,NNCN
      NCP(NCN(J))=J+NN+NN
  107 CONTINUE

 READ SYSTEM TOPOLOGY (ELEMENT NO. AND NODE NUMBERS IN
 COUNTERCLOCKWISE FASHION STARTING AT THE UPPER LEFT
 HAND CORNER NODE).

      DO 105 I=1,NE
      READ(NREAD,1010)J,NODE(J,1),NODE(J,2),NODE(J,3),
     1NODE(J,4),NODE(J,5),NODE(J,6)
  105 CONTINUE
      MAXDIF=0
      DO 108 I=1,NE
      DO 108 J=1,6
      DO 108 K=1,6
      LL=IABS(NODE(I,J)-NODE(I,K))
      IF(LL.GT.MAXDIF) MAXDIF=LL
      IBAND=2*(MAXDIF+1)
      NEQ=3*NN+NNCN
 108  CONTINUE
      WRITE(NWRITE,1017)IBAND,NEQ
 1017 FORMAT(5X,'IBAND=',I3,//,5X,'NEQ =',I3,//)

 READ NODES WHERE BOTH U AND V VELOCITY IS SPECIFIED

      DO 110 I=1,MM
      READ(NREAD,1006)WORD,NVELS,VELU,VELV
      IF(WORD.EQ.STOP) GO TO 111
      NVS(I)=NVELS
      X(NVS(I))=VELU
      X(NVS(I)+NN=VELV
  110 CONTINUE
```

```
COUNT NODES HAVING SPECIFIED VELOCITIES

  111 NNVELS=I-1

READ QX AND QY VALUES AT INTERNAL NODES

      DO 125 I=1,NN
      READ(NREAD,1006)WORD,NQXY,QXNS,QYNS
      IF(WORD.EQ.STOP) GO TO 126
      NCS(I)=NQXY
      Q(NQS(I))=QXNS
      Q(NQS(I)+NN)=QYNS
  125 CONTINUE

COUNT NODES HAVING SPECIFIED QX AND QY

  126 NNQXY=I-1

READ NODE NUMBER AND PRESSURE WHERE SPECIFIED

      DO 130 I=1,NN
      READ(NREAD,1025)WORD,NP,PNP
      IF(WORD.EQ.STOP)GO TO 135
      NPS(I)=NP
      X(NCP(NPS(I)))=PNP
  130 CONTINUE

COUNT BOUNDARY NODES WHERE PRESSURE IS SPECIFIED

  135 NNPS=I-1

READ NODE NUMBER AND TEMPERATURE WHERE SPECIFIED

      DO 140 I=1,MM
      READ(NREAD,1025) WORD,NTEMP,TNT
      IF(WORD.EQ.STOP) GO TO  145
      NVS(I+ NNVELS)=NTEMP
      X(NVS(I+NNVELS)+MM)=TNT
  140 CONTINUE

COUNT NODES HAVING SPECIFIED TEMPERATURES

  145 NNTS=I-1

READ NODE NUMBERS AND QZC WHERE SPECIFIED

      DO 141 I=1,MM
```

```
      READ(NREAD,1025) WORD,NQZC,QZCNS
      IF(WORD.EQ.STOP) GO TO 146
      NQS(NNQXY+I)=NQZC
      Q(NQS(NNQXY+I)+2*NN)=QZCNS
  141 CONTINUE

COUNT NODES WHERE QZC IS SPECIFIED

  146 NNQZC=I-1

READ NODE NUMBERS AND HEAT FLUX QZ WHERE SPECIFIED

      DO 142 I=1,MM
      READ(NREAD,1025) WORD,NQZ,QZNS
      IF(WORD.EQ.STOP) GO TO  147
      NQS(NNQXY+NNQZC+I)=NQZ
      Q(NQS(NNQXY+NNQZC+I)+MM)=QZNS
  142 CONTINUE

COUNT NODES WHERE HEAT FLUX QZ IS SPECIFIED

  147 NNQZ=I-1

NQIS IS A LIST OF INDICES OF KNOWN QX,QY,QZC AND QZ

      DO 1140 I=1,NNQXY
      NQIS(I)=NQS(I)
      NQIS(I+NNQXY)=NQS(I)+NN
 1140 CONTINUE
      DO 1141 I=1,NNQZC
      NQIS(2*NNQXY+I)=NQS(NNQXY+I)+2*NN
 1141 CONTINUE
      DO 1145 I=1,NNQZ
      NQIS(2*NNQXY+NNQZC+I)=NQS(NNQXY+NNQZC+I)+MM
 1145 CONTINUE

NVIS IS A LIST OF KNOWN VELOCITY, PRESSURE, AND TEMPERATURE INDICES

      DO 1150 I=1,NNVELS
      NVIS(I)=NVS(I)
      NVIS(I+NNVELS)=NVS(I)+NN
 1150 CONTINUE
      DO 1155 J=1,NNPS
      NVIS(2*NNVELS+J)=NCP(NPS(J))
 1155 CONTINUE
      DO 1160 K=1,NNTS
      NVIS(2*NNVELS+NNPS+K)=NVS(K+NNVELS)+MM
 1160 CONTINUE
```

```
NNHC=NUMBER OF NODES WHERE HEAT TRANSFER COEFFICIENT IS SPECIFIED

      NNHC=0

NTOTQ=TOTAL NUMBER OF KNOWN QX,QY,QZC, AND QZ

      NTOTQ=2*NNQXY+NNQZC+NNQZ

NTOTVP=TOTAL NUMBER OF KNOWN VELOCITIES, PRESSURES, AND TEMPERATURES

      NTOTVP=2*NNVELS+NNPS+NNTS

PRINT ALL INPUT DATA

      WRITE(NWRITE,1035)NN,NE,NNCN
      WRITE(NWRITE,1036)NNVELS
      WRITE(NWRITE,1037)NNQXY
      WRITE(NWRITE,1038)NNPS
      WRITE(NWRITE,1039)NNTS
      WRITE(NWRITE,1034)NNQZC
      WRITE(NWRITE,1040)NNQZ
      WRITE(NWRITE,1041)
      DO 150 I=1,NNCN
      WRITE(NWRITE,1045)NCN(I),XC(NCN(I)),YC(NCN(I))
  150 CONTINUE
      WRITE(NWRITE,1050)
      DO 155 I=1,NE
      WRITE(NWRITE,1055)I,NODE(I,1),NODE(I,2),NODE(I,3),
     1NODE(I,4),NODE(I,5),NODE(I,6)
  155 CONTINUE
      WRITE(NWRITE,1060)
      DO 160 I=1,NNVELS
      WRITE(NWRITE,1065)I,NVS(I),X(NVS(I)),X(NVS(I)+NN)
  160 CONTINUE
      WRITE(NWRITE,1070)
      DO 165 I=1,NNQXY
      WRITE(NWRITE,1065)I,NQS(I),Q(NQS(I)),Q(NQS(I)+NN)
  165 CONTINUE
      WRITE(NWRITE,1080)
      DO 170 I=1,NNPS
      WRITE(NWRITE,1085)I,NPS(I),X(NCP(NPS(I)))
  170 CONTINUE
      WRITE(NWRITE,1081)
      DO 171 I=1,NNTS
      WRITE(NWRITE,1085)I,NVS(I+NNVELS),X(NVS(I+NNVELS)+MM)
  171 CONTINUE
      WRITE(NWRITE,1083)
```

```
      DO 173 I=1,NNQZC                                                  TIME0279
      WRITE(NWRITE,1085)I,NQS(I+NNQXY),Q(NQS(I+NNQXY)+2*NN)             TIME0280
  173 CONTINUE                                                          TIME0281
      WRITE(NWRITE,1082)                                                TIME0282
      DO 172 I=1,NNQZ                                                   TIME0283
      WRITE(NWRITE,1085)I,NQS(I+NNQXY+NNQZC),Q(NQS(I+NNQXY+NNQZC)+MM)   TIME0284
  172 CONTINUE                                                          TIME0285
      T=0.D0                                                            TIME0286
  177 CONTINUE                                                          TIME0287
      DO 178 I=1,MMM                                                    TIME0288
      RHS(I)=0.D0                                                       TIME0289
      DO 178 J=1,MMM                                                    TIME0290
      TM(I,J)=0.D0                                                      TIME0291
  178 CONTINUE                                                          TIME0292
      IF(T.GT.0.D0) GO TO 180                                           TIME0293
      DO 179 I=1,MMM                                                    TIME0294
      DC 179 J=1,MMM                                                    TIME0295
      CC(I,J)=0.D0                                                      TIME0296
  179 CONTINUE                                                          TIME0297
  180 CONTINUE                                                          TIME0298
      DO 181 I=1,21                                                     TIME0299
      DO 181 J=1,21                                                     TIME0300
      TM$(I,J)=0.D0                                                     TIME0301
      CC$(I,J)=0.D0                                                     TIME0302
  181 CONTINUE                                                          TIME0303
                                                                        TIME0304
 END OF INPUT AND VERIFICATION ROUTINE                                  TIME0305
                                                                        TIME0306
      DO 300 K=1,NE                                                     TIME0307
      N1=NODE(K,1)                                                      TIME0308
      N2=NODE(K,2)                                                      TIME0309
      N3=NODE(K,3)                                                      TIME0310
      N4=NODE(K,4)                                                      TIME0311
      N5=NODE(K,5)                                                      TIME0312
      N6=NODE(K,6)                                                      TIME0313
      N7=NODE(K,1)+NN                                                   TIME0314
      N8=NODE(K,2)+NN                                                   TIME0315
      N9=NODE(K,3)+NN                                                   TIME0316
      N10=NODE(K,4)+NN                                                  TIME0317
      N11=NODE(K,5)+NN                                                  TIME0318
      N12=NODE(K,6)+NN                                                  TIME0319
      N13=NCP(NODE(K,1))                                                TIME0320
      N14=NCP(NODE(K,3))                                                TIME0321
      N15=NCP(NODE(K,5))                                                TIME0322
      N16=NODE(K,1)+MM                                                  TIME0323
      N17=NODE(K,2)+MM                                                  TIME0324
      N18=NODE(K,3)+MM                                                  TIME0325
      N19=NODE(K,4)+MM                                                  TIME0326
```

```
      N20=NODE(K,5)+MM                                                  TIME0327
      N21=NODE(K,6)+MM                                                  TIME0328
      XC$(1)=XC(NODE(K,1))                                              TIME0329
      XC$(2)=XC(NODE(K,3))                                              TIME0330
      XC$(3)=XC(NODE(K,5))                                              TIME0331
      YC$(1)=YC(NODE(K,1))                                              TIME0332
      YC$(2)=YC(NODE(K,3))                                              TIME0333
      YC$(3)=YC(NODE(K,5))                                              TIME0334
      A=1.DO                                                            TIME0335
      ZBAR=(YC$(1)+YC$(2)+YC$(3))/3.DO                                  TIME0336
      RBAR=(XC$(1)+XC$(2)+XC$(3))/3.DO                                  TIME0337
      IF(NCASE.EQ.2)A=RBAR                                              TIME0338
      AA=1.DO                                                           TIME0339
      IF(NCASE.EQ.2)AA=2.DO*3.14159DO*(RBAR)                            TIME0340
      A1=XC$(2)*YC$(3)-XC$(3)*YC$(2)                                    TIME0341
      A2=XC$(3)*YC$(1)-XC$(1)*YC$(3)                                    TIME0342
      A3=XC$(1)*YC$(2)-XC$(2)*YC$(1)                                    TIME0343
      B1=YC$(2)-YC$(3)                                                  TIME0344
      B2=YC$(3)-YC$(1)                                                  TIME0345
      B3=YC$(1)-YC$(2)                                                  TIME0346
      C1=XC$(3)-XC$(2)                                                  TIME0347
      C2=XC$(1)-XC$(3)                                                  TIME0348
      C3=XC$(2)-XC$(1)                                                  TIME0349
      DEL=DABS(0.5DO*(XC$(1)*(YC$(2)-YC$(3))+XC$(2)*(YC$(3)-YC$(1))     TIME0350
     1+XC$(3)*(YC$(1)-YC$(2))))                                         TIME0351
      CONST=10.90DO*(1.DO*A/(3.DO*DEL))*AA                              TIME0352
      D1=-B1/6.DO                                                       TIME0353
      D2=-B2/6.DO                                                       TIME0354
      D3=-B3/6.DO                                                       TIME0355
      E1=-C1/6.DO                                                       TIME0356
      E2=-C2/6.DO                                                       TIME0357
      E3=-C3/6.DO                                                       TIME0358
      F1=B1/2520.DO                                                     TIME0359
      F2=B2/2520.DO                                                     TIME0360
      F3=B3/2520.DO                                                     TIME0361
      G1=C1/2520.DO                                                     TIME0362
      G2=C2/2520.DO                                                     TIME0363
      G3=C3/2520.DO                                                     TIME0364
      U1=T1(N1)                                                         TIME0365
      U2=T1(N2)                                                         TIME0366
      U3=T1(N3)                                                         TIME0367
      U4=T1(N4)                                                         TIME0368
      U5=T1(N5)                                                         TIME0369
      U6=T1(N6)                                                         TIME0370
      V1=T1(N7)                                                         TIME0371
      V2=T1(N8)                                                         TIME0372
      V3=T1(N9)                                                         TIME0373
      V4=T1(N10)                                                        TIME0374
```

```
      V5=T1(N11)                                                        TIME0375
      V6=T1(N12)                                                        TIME0376
      TM$(1,1)=0.75D0*(B1*B1+C1*C1)*CONST                               TIME0377
      TM$(1,2)=(B1*B2+C1*C2)*CONST                                      TIME0378
      TM$(1,3)=-TM$(1,2)*0.25D0                                         TIME0379
      TM$(1,4)=0.                                                       TIME0380
      TM$(1,6)=(B1*B3+C1*C3)*CONST                                      TIME0381
      TM$(1,5)=-TM$(1,6)*.25D0                                          TIME0382
      TM$(3,3)=.75D0*(B2*B2+C2*C2)*CONST                                TIME0383
      TM$(3,4)=(B2*B3+C2*C3)*CONST                                      TIME0384
      TM$(3,5)=-TM$(3,4)*0.25D0                                         TIME0385
      TM$(2,1)=TM$(1,2)                                                 TIME0386
      TM$(2,3)=TM$(1,2)                                                 TIME0387
      TM$(2,2)=8.D0/3.D0*(TM$(1,1)+TM$(3,3))+2.D0*TM$(1,2)              TIME0388
      TM$(2,4)=2.D0*TM$(1,6)+TM$(3,4)+TM$(1,2)+4.D0/3.D0*TM$(3,3)       TIME0389
      TM$(2,5)=0.                                                       TIME0390
      TM$(2,6)=TM$(1,6)+2.D0*TM$(3,4)+TM$(1,2)+4.D0/3.D0*TM$(1,1)       TIME0391
      TM$(3,1)=TM$(1,3)                                                 TIME0392
      TM$(3,2)=TM$(2,3)                                                 TIME0393
      TM$(3,6)=0.                                                       TIME0394
      TM$(5,5)=.75D0*(B3*B3+C3*C3)*CONST                                TIME0395
      TM$(4,1)=TM$(1,4)                                                 TIME0396
      TM$(4,2)=TM$(2,4)                                                 TIME0397
      TM$(4,3)=TM$(3,4)                                                 TIME0398
      TM$(4,4)=8.D0/3.D0*(TM$(3,3)+TM$(5,5))+2.D0*TM$(3,4)              TIME0399
      TM$(4,5)=TM$(3,4)                                                 TIME0400
      TM$(4,6)=TM$(1,6)+TM$(3,4)+2.D0*TM$(1,2)+4.D0/3.D0*TM$(5,5)       TIME0401
      TM$(6,3)=TM$(3,6)                                                 TIME0402
      TM$(5,1)=TM$(1,5)                                                 TIME0403
      TM$(5,2)=TM$(2,5)                                                 TIME0404
      TM$(5,3)=TM$(3,5)                                                 TIME0405
      TM$(5,4)=TM$(4,5)                                                 TIME0406
      TM$(5,6)=TM$(1,6)                                                 TIME0407
      TM$(6,1)=TM$(1,6)                                                 TIME0408
      TM$(6,2)=TM$(2,6)                                                 TIME0409
      TM$(6,4)=TM$(4,6)                                                 TIME0410
      TM$(6,5)=TM$(5,6)                                                 TIME0411
      TM$(6,6)=8.D0/3.D0*(TM$(5,5)+TM$(1,1))+2.D0*TM$(1,6)              TIME0412
      IF(NCASE.NE.1) GO TO 3000                                         TIME0413
                                                                        TIME0414
 BEGIN INPUT OF NON-LINEAR TERMS                                        TIME0415
                                                                        TIME0416
      TM$(1,1)=TM$(1,1)                                                 TIME0417
     1-(78.D0*U1+48.D0*U2-9.D0*U3+12.D0*U4-9.D0*U5+48.D0*U6)*F1         TIME0418
     2             -(78.D0*V1+48.D0*V2-9.D0*V3+12.D0*V4-9.D0*V5+48.D0*V6)TIME0419
     3*G1                                                               TIME0420
      TM$(2,1)=TM$(2,1)                                                 TIME0421
     1-(48.D0*U1+160.D0*U2-32.D0*U3+16.D0*U4-20.D0*U5+80.D0*U6)*F1      TIME0422
```

```
2           -(48.D0*V1+160.D0*V2-32.D0*V3+16.D0*V4-20.D0*V5+80.D0*TIME0423
3V6)*G1                                                                TIME0424
 TM$(3,1)=TM$(3,1)                                                     TIME0425
1-(-9.D0*U1-32.D0*U2-18.D0*U3-16.D0*U4+11.D0*U5-20.D0*U6)*F1           TIME0426
2           -(-9.D0*V1-32.D0*V2-18.D0*V3-16.D0*V4+11.D0*V5-20.D0*VTIME0427
36)*G1                                                                 TIME0428
 TM$(4,1)=TM$(4,1)                                                     TIME0429
1-(12.D0*U1+16.D0*U2-16.D0*U3-96.D0*U4-16.D0*U5+16.D0*U6)*F1           TIME0430
2           -(12.D0*V1+16.D0*V2-16.D0*V3-96.D0*V4-16.D0*V5+16.D0*VTIME0431
36)*G1                                                                 TIME0432
 TM$(5,1)=TM$(5,1)                                                     TIME0433
1-(-9.D0*U1-20.D0*U2+11.D0*U3-16.D0*U4-18.D0*U5-32.D0*U6)*F1           TIME0434
2           -(-9.D0*V1-20.D0*V2+11.D0*V3-16.D0*V4-18.D0*V5-32.D0*VTIME0435
36)*G1                                                                 TIME0436
 TM$(6,1)=TM$(6,1)                                                     TIME0437
1-(48.D0*U1+80.D0*U2-20.D0*U3+16.D0*U4-32.D0*U5+160.D0*U6)*F1          TIME0438
2           -(48.D0*V1+80.D0*V2-20.D0*V3+16.D0*V4-32.D0*V5+160.D0*TIME0439
3V6)*G1                                                                TIME0440
 TM$(1,2)=TM$(1,2)                                                     TIME0441
1-(24.D0*U1-32.D0*U2-16.D0*U3-48.D0*U4+4.D0*U5-16.D0*U6)*F1            TIME0442
2           -(24.D0*V1-32.D0*V2-16.D0*V3-48.D0*V4+4.D0*V5-16.D0*V6TIME0443
3)*G1                 -(120.D0*U1+48.D0*U2-16.D0*U3-16.D0*U4-16.TIME0444
4D0*U5+48.D0*U6)*F2                 -(120.D0*V1+48.D0*V2-16.D0*V3-TIME0445
516.D0*V4-16.D0*V5+48.D0*V6)*G2                                        TIME0446
 TM$(2,2)=TM$(2,2)                                                     TIME0447
1-(-32.D0*U1+384.D0*U2+48.D0*U3+192.D0*U4-48.D0*U5+128.D0*U6)*F1       TIME0448
2           -(-32.D0*V1+384.D0*V2+48.D0*V3+192.D0*V4-48.D0*V5+128.TIME0449
3D0*V6)*G1            -(48.D0*U1+384.D0*U2-32.D0*U3+128.D0*U4-48TIME0450
4.D0*U5+192.D0*U6)*F2               -(48.D0*V1+384.D0*V2-32.D0*V3+TIME0451
5128.D0*V4-48.D0*V5+192.D0*V6)*G2                                      TIME0452
 TM$(3,2)=TM$(3,2)                                                     TIME0453
1-(-16.D0*U1+48.D0*U2+120.D0*U3+48.D0*U4-16.D0*U5-16.D0*U6)*F1         TIME0454
2           -(-16.D0*V1+48.D0*V2+120.D0*V3+48.D0*V4-16.D0*V5-16.D0TIME0455
3*V6)*G1              -(-16.D0*U1-32.D0*U2+24.D0*U3-16.D0*U4+4.DTIME0456
40*U5-48.D0*U6)*F2                  -(-16.D0*V1-32.D0*V2+24.D0*V3-TIME0457
516.D0*V4+4.D0*V5-48.D0*V6)*G2                                         TIME0458
 TM$(4,2)=TM$(4,2)                                                     TIME0459
1-(-48.D0*U1+192.D0*U2+48.D0*U3+384.D0*U4-32.D0*U5+128.D0*U6)*F1       TIME0460
2           -(-48.D0*V1+192.D0*V2+48.D0*V3+384.D0*V4-32.D0*V5+128.TIME0461
3D0*V6)*G1            -(-16.D0*U1+128.D0*U2-16.D0*U3+128.D0*U4-1TIME0462
46.D0*U5+128.D0*U6)*F2              -(-16.D0*V1+128.D0*V2-16.D0*V3TIME0463
5+128.D0*V4-16.D0*V5+128.D0*V6)*G2                                     TIME0464
 TM$(5,2)=TM$(5,2)                                                     TIME0465
1-(4.D0*U1-48.D0*U2-16.D0*U3-32.D0*U4+24.D0*U5-16.D0*U6)*F1            TIME0466
2           -(4.D0*V1-48.D0*V2-16.D0*V3-32.D0*V4+24.D0*V5-16.D0*V6TIME0467
3)*G1                 -(-16.D0*U1-48.D0*U2+4.D0*U3-16.D0*U4+24.DTIME0468
40*U5-32.D0*U6)*F2                  -(-16.D0*V1-48.D0*V2+4.D0*V3-1TIME0469
56.D0*V4+24.D0*V5-32.D0*V6)*G2                                         TIME0470
```

```
      TM$(6,2)=TM$(6,2)                                                TIME0471
     1-(-16.D0*U1+128.D0*U2-16.D0*U3+128.D0*U4-16.D0*U5+128.D0*U6)*F1  TIME0472
     2               -(-16.D0*V1+128.D0*V2-16.D0*V3+128.D0*V4-16.D0*V5+128.TIME0473
     3D0*V6)*G1                  -(48.D0*U1+192.D0*U2-48.D0*U3+128.D0*U4-32TIME0474
     4.D0*U5+384.D0*U6)*F2                  -(48.D0*V1+192.D0*V2-48.D0*V3+TIME0475
     5128.D0*V4-32.D0*V5+384.D0*V6)*G2                                 TIME0476
      TM$(1,3)=TM$(1,3)                                                TIME0477
     1-(-18.D0*U1-32.D0*U2-9.D0*U3-20.D0*U4+11.D0*U5-16.D0*U6)*F2      TIME0478
     2               -(-18.D0*V1-32.D0*V2-9.D0*V3-20.D0*V4+11.D0*V5-16.D0*VTIME0479
     36)*G2                                                            TIME0480
      TM$(2,3)=TM$(2,3)                                                TIME0481
     1-(-32.D0*U1+160.D0*U2+48.D0*U3+80.D0*U4-20.D0*U5+16.D0*U6)*F2    TIME0482
     2               -(-32.D0*V1+160.D0*V2+48.D0*V3+80.D0*V4-20.D0*V5+16.D0TIME0483
     3*V6)*G2                                                          TIME0484
      TM$(3,3)=TM$(3,3)                                                TIME0485
     1-(-9.D0*U1+48.D0*U2+78.D0*U3+48.D0*U4-9.D0*U5+12.D0*U6)*F2       TIME0486
     2               -(-9.D0*V1+48.D0*V2+78.D0*V3+48.D0*V4-9.D0*V5+12.D0*V6TIME0487
     3)*G2                                                             TIME0488
      TM$(4,3)=TM$(4,3)                                                TIME0489
     1-(-20.D0*U1+80.D0*U2+48.D0*U3+160.D0*U4-32.D0*U5+16.D0*U6)*F2    TIME0490
     2               -(-20.D0*V1+80.D0*V2+48.D0*V3+160.D0*V4-32.D0*V5+16.D0TIME0491
     3*V6)*G2                                                          TIME0492
      TM$(5,3)=TM$(5,3)                                                TIME0493
     1-(11.D0*U1-20.D0*U2-9.D0*U3-32.D0*U4-18.D0*U5-16.D0*U6)*F2       TIME0494
     2               -(11.D0*V1-20.D0*V2-9.D0*V3-32.D0*V4-18.D0*V5-16.D0*V6TIME0495
     3)*G2                                                             TIME0496
      TM$(6,3)=TM$(6,3)                                                TIME0497
     1-(-16.D0*U1+16.D0*U2+12.D0*U3+16.D0*U4-16.D0*U5-96.D0*U6)*F2     TIME0498
     2               -(-16.D0*V1+16.D0*V2+12.D0*V3+16.D0*V4-16.D0*V5-96.D0*TIME0499
     3V6)*G2                                                           TIME0500
      TM$(1,4)=TM$(1,4)                                                TIME0501
     1-(24.D0*U1-16.D0*U2+4.D0*U3-48.D0*U4-16.D0*U5-32.D0*U6)*F2       TIME0502
     2               -(24.D0*V1-16.D0*V2+4.D0*V3-48.D0*V4-16.D0*V5-32.D0*V6TIME0503
     3)*G2                       -(24.D0*U1-32.D0*U2-16.D0*U3-48.D0*U4+4.D0TIME0504
     4*U5-16.D0*U6)*F3                      -(24.D0*V1-32.D0*V2-16.D0*V3-4TIME0505
     58.D0*V4+4.D0*V5-16.D0*V6)*G3                                     TIME0506
      TM$(2,4)=TM$(2,4)                                                TIME0507
     1-(-16.D0*U1+128.D0*U2-16.D0*U3+128.D0*U4-16.D0*U5+128.D0*U6)*F2  TIME0508
     2               -(-16.D0*V1+128.D0*V2-16.D0*V3+128.D0*V4-16.D0*V5+128.TIME0509
     3D0*V6)*G2                  -(-32.D0*U1+384.D0*U2+48.D0*U3+192.D0*U4-4TIME0510
     48.D0*U5+128.D0*U6)*F3                 -(-32.D0*V1+384.D0*V2+48.D0*V3TIME0511
     5+192.D0*V4-48.D0*V5+128.D0*V6)*G3                                TIME0512
      TM$(3,4)=TM$(3,4)                                                TIME0513
     1-(4.D0*U1-16.D0*U2+24.D0*U3-32.D0*U4+16.D0*U5-48.D0*U6)*F2       TIME0514
     2               -(4.D0*V1-16.D0*V2+24.D0*V3-32.D0*V4+16.D0*V5-48.D0*V6TIME0515
     3)*G2                       -(-16.D0*U1+48.D0*U2+120.D0*U3+48.D0*U4-16TIME0516
     4.D0*U5-16.D0*U6)*F3                   -(-16.D0*V1+48.D0*V2+120.D0*V3TIME0517
     5+48.D0*V4-16.D0*V5-16.D0*V6)*G3                                  TIME0518
```

```
      TM$(4,4)=TM$(4,4)                                                 TIME0519
     1-(-48.D0*U1+128.D0*U2-32.D0*U3+384.D0*U4+48.D0*U5+192.D0*U6)*F2   TIME0520
     2              -(-48.D0*V1+128.D0*V2-32.D0*V3+384.D0*V4+48.D0*V5+192.TIME0521
     3D0*V6)*G2                -(-48.D0*U1+192.D0*U2+48.D0*U3+384.D0*U4-3TIME0522
     42.D0*U5+128.D0*U6)*F3                -(-48.D0*V1+192.D0*V2+48.D0*V3TIME0523
     5+384.D0*V4-32.D0*V5+128.D0*V6)*G3                                 TIME0524
      TM$(5,4)=TM$(5,4)                                                 TIME0525
     1-(-16.D0*U1-16.D0*U2-16.D0*U3+48.D0*U4+120.D0*U5+48.D0*U6)*F2     TIME0526
     2              -(-16.D0*V1-16.D0*V2-16.D0*V3+48.D0*V4+120.D0*V5+48.D0TIME0527
     3*V6)*G2                  -(4.D0*U1-48.D0*U2-16.D0*U3-32.D0*U4+24.D0TIME0528
     4*U5-16.D0*U6)*F3                     -(4.D0*V1-48.D0*V2-16.D0*V3-32TIME0529
     5.D0*V4+24.D0*V5-16.D0*V6)*G3                                      TIME0530
      TM$(6,4)=TM$(6,4)                                                 TIME0531
     1-(-32.D0*U1+128.D0*U2-48.D0*U3+192.D0*U4+48.D0*U5+384.D0*U6)*F2   TIME0532
     2              -(-32.D0*V1+128.D0*V2-48.D0*V3+192.D0*V4+48.D0*V5+384.TIME0533
     3D0*V6)*G2                -(-16.D0*U1+128.D0*U2-16.D0*U3+128.D0*U4-1TIME0534
     46.D0*U5+128.D0*U6)*F3                -(-16.D0*V1+128.D0*V2-16.D0*V3TIME0535
     5+128.D0*V4-16.D0*V5+128.D0*V6)*G3                                 TIME0536
      TM$(1,5)=TM$(1,5)                                                 TIME0537
     1-(-18.D0*U1-16.D0*U2+11.D0*U3-20.D0*U4-9.D0*U5-32.D0*U6)*F3       TIME0538
     2              -(-18.D0*V1-16.D0*V2+11.D0*V3-20.D0*V4-9.D0*V5-32.D0*VTIME0539
     36)*G3                                                             TIME0540
      TM$(2,5)=TM$(2,5)                                                 TIME0541
     1-(-16.D0*U1-96.D0*U2-16.D0*U3+16.D0*U4+12.D0*U5+16.D0*U6)*F3      TIME0542
     2              -(-16.D0*V1-96.D0*V2-16.D0*V3+16.D0*V4+12.D0*V5+16.D0*TIME0543
     3V6)*G3                                                            TIME0544
      TM$(3,5)=TM$(3,5)                                                 TIME0545
     1-(11.D0*U1-16.D0*U2-18.D0*U3-32.D0*U4-9.D0*U5-20.D0*U6)*F3        TIME0546
     2              -(11.D0*V1-16.D0*V2-18.D0*V3-32.D0*V4-9.D0*V5-20.D0*V6TIME0547
     3)*G3                                                              TIME0548
      TM$(4,5)=TM$(4,5)                                                 TIME0549
     1-(-20.D0*U1+16.D0*U2-32.D0*U3+160.D0*U4+48.D0*U5+80.D0*U6)*F3     TIME0550
     2              -(-20.D0*V1+16.D0*V2-32.D0*V3+160.D0*V4+48.D0*V5+80.D0TIME0551
     3*V6)*G3                                                           TIME0552
      TM$(5,5)=TM$(5,5)                                                 TIME0553
     1-(-9.D0*U1+12.D0*U2-9.D0*U3+48.D0*U4+78.D0*U5+48.D0*U6)*F3        TIME0554
     2              -(-9.D0*V1+12.D0*V2-9.D0*V3+48.D0*V4+78.D0*V5+48.D0*V6TIME0555
     3)*G3                                                              TIME0556
      TM$(6,5)=TM$(6,5)                                                 TIME0557
     1-(-32.D0*U1+16.D0*U2-20.D0*U3+80.D0*U4+48.D0*U5+160.D0*U6)*F3     TIME0558
     2              -(-32.D0*V1+16.D0*V2-20.D0*V3+80.D0*V4+48.D0*V5+160.D0TIME0559
     3*V6)*G3                                                           TIME0560
      TM$(1,6)=TM$(1,6)                                                 TIME0561
     1-(24.D0*U1-16.D0*U2+4.D0*U3-48.D0*U4-16.D0*U5-32.D0*U6)*F1        TIME0562
     2              -(24.D0*V1-16.D0*V2+4.D0*V3-48.D0*V4-16.D0*V5-32.D0*V6TIME0563
     3)*G1                     -(120.D0*U1+48.D0*U2-16.D0*U3-16.D0*U4-16.TIME0564
     4D0*U5+48.D0*U6)*F3                   -(120.D0*V1+48.D0*V2-16.D0*V3-TIME0565
     516.D0*V4-16.D0*V5+48.D0*V6)*G3                                    TIME0566
```

```
      TM$(2,6)=TM$(2,6)
     1-(-16.D0*U1+128.D0*U2-16.D0*U3+128.D0*U4-16.D0*U5+128.D0*U6)*F1
     2              -(-16.D0*V1+128.D0*V2-16.D0*V3+128.D0*V4-16.D0*V5+128.
     3D0*V6)*G1                -(48.D0*U1+384.D0*U2-32.D0*U3+128.D0*U4-48
     4.D0*U5+192.D0*U6)*F3                    -(48.D0*V1+384.D0*V2-32.D0*V3+
     5128.D0*V4-48.D0*V5+192.D0*V6)*G3
      TM$(3,6)=TM$(3,6)
     1-(4.D0*U1-16.D0*U2+24.D0*U3-32.D0*U4-16.D0*U5-48.D0*U6)*F1
     2              -(4.D0*V1-16.D0*V2+24.D0*V3-32.D0*V4-16.D0*V5-48.D0*V6
     3)*G1                     -(-16.D0*U1-32.D0*U2+24.D0*U3-16.D0*U4+4.D
     40*U5-48.D0*U6)*F3                       -(-16.D0*V1-32.D0*V2+24.D0*V3-
     516.D0*V4+4.D0*V5-48.D0*V6)*G3
      TM$(4,6)=TM$(4,6)
     1-(-48.D0*U1+128.D0*U2-32.D0*U3+384.D0*U4+48.D0*U5+192.D0*U6)*F1
     2              -(-48.D0*V1+128.D0*V2-32.D0*V3+384.D0*V4+48.D0*V5+192.
     3D0*V6)*G1                -(-16.D0*U1+128.D0*U2-16.D0*U3+128.D0*U4-1
     46.D0*U5+128.D0*U6)*F3                   -(-16.D0*V1+128.D0*V2-16.D0*V3
     5+128.D0*V4-16.D0*V5+128.D0*V6)*G3
      TM$(5,6)=TM$(5,6)
     1-(-16.D0*U1-16.D0*U2-16.D0*U3+48.D0*U4+120.D0*U5+48.D0*U6)*F1
     2              -(-16.D0*V1-16.D0*V2-16.D0*V3+48.D0*V4+120.D0*V5+48.D0
     3*V6)*G1                  -(-16.D0*U1-48.D0*U2+4.D0*U3-16.D0*U4+24.D
     40*U5-32.D0*U6)*F3                       -(-16.D0*V1-48.D0*V2+4.D0*V3-1
     56.D0*V4+24.D0*V5-32.D0*V6)*G3
      TM$(6,6)=TM$(6,6)
     1-(-32.D0*U1+128.D0*U2-48.D0*U3+192.D0*U4+48.D0*U5+384.D0*U6)*F1
     2              -(-32.D0*V1+128.D0*V2-48.D0*V3+192.D0*V4+48.D0*V5+384.
     3D0*V6)*G1                -(48.D0*U1+192.D0*U2-48.D0*U3+128.D0*U4-32
     4.D0*U5+384.D0*U6)*F3                    -(48.D0*V1+192.D0*V2-48.D0*V3+
     5128.D0*V4-32.D0*V5+384.D0*V6)*G3

C THIS ENDS ADDITION OF NON-LINEAR TERMS TO THE LOCAL ARRAY

 3000 CONTINUE
      TM$(7,7)=TM$(1,1)
      TM$(7,8)=TM$(1,2)
      TM$(7,9)=TM$(1,3)
      TM$(7,10)=TM$(1,4)
      TM$(7,11)=TM$(1,5)
      TM$(7,12)=TM$(1,6)
      TM$(8,7)=TM$(2,1)
      TM$(8,8)=TM$(2,2)
      TM$(8,9)=TM$(2,3)
      TM$(8,10)=TM$(2,4)
      TM$(8,11)=TM$(2,5)
      TM$(8,12)=TM$(2,6)
      TM$(9,7)=TM$(3,1)
      TM$(9,8)=TM$(3,2)
```

```
      TM$(9,9)=TM$(3,3)                                                 TIME0615
      TM$(9,10)=TM$(3,4)                                                TIME0616
      TM$(9,11)=TM$(3,5)                                                TIME0617
      TM$(9,12)=TM$(3,6)                                                TIME0618
      TM$(10,7)=TM$(4,1)                                                TIME0619
      TM$(10,8)=TM$(4,2)                                                TIME0620
      TM$(10,9)=TM$(4,3)                                                TIME0621
      TM$(10,10)=TM$(4,4)                                               TIME0622
      TM$(10,11)=TM$(4,5)                                               TIME0623
      TM$(10,12)=TM$(4,6)                                               TIME0624
      TM$(11,7)=TM$(5,1)                                                TIME0625
      TM$(11,8)=TM$(5,2)                                                TIME0626
      TM$(11,9)=TM$(5,3)                                                TIME0627
      TM$(11,10)=TM$(5,4)                                               TIME0628
      TM$(11,11)=TM$(5,5)                                               TIME0629
      TM$(11,12)=TM$(5,6)                                               TIME0630
      TM$(12,7)=TM$(6,1)                                                TIME0631
      TM$(12,8)=TM$(6,2)                                                TIME0632
      TM$(12,9)=TM$(6,3)                                                TIME0633
      TM$(12,10)=TM$(6,4)                                               TIME0634
      TM$(12,11)=TM$(6,5)                                               TIME0635
      TM$(12,12)=TM$(6,6)                                               TIME0636
      TM$(13,1)=D1                                                      TIME0637
      TM$(14,1)=0.D0                                                    TIME0638
      TM$(15,1)=0.D0                                                    TIME0639
      TM$(13,2)=D1+2.D0*D2                                              TIME0640
      TM$(14,2)=2.D0*D1+D2                                              TIME0641
      TM$(15,2)=D1+D2                                                   TIME0642
      TM$(13,3)=0.D0                                                    TIME0643
      TM$(14,3)=D2                                                      TIME0644
      TM$(15,3)=0.D0                                                    TIME0645
      TM$(13,4)=D2+D3                                                   TIME0646
      TM$(14,4)=D2+2.D0*D3                                              TIME0647
      TM$(15,4)=2.D0*D2+D3                                              TIME0648
      TM$(13,5)=0.D0                                                    TIME0649
      TM$(14,5)=0.D0                                                    TIME0650
      TM$(15,5)=D3                                                      TIME0651
      TM$(13,6)=D1+2.D0*D3                                              TIME0652
      TM$(14,6)=D1+D3                                                   TIME0653
      TM$(15,6)=2.D0*D1+D3                                              TIME0654
      TM$(13,7)=E1                                                      TIME0655
      TM$(14,7)=0.D0                                                    TIME0656
      TM$(15,7)=0.D0                                                    TIME0657
      TM$(13,8)=E1+2.D0*E2                                              TIME0658
      TM$(14,8)=2.D0*E1+E2                                              TIME0659
      TM$(15,8)=E1+E2                                                   TIME0660
      TM$(13,9)=0.D0                                                    TIME0661
      TM$(14,9)=E2                                                      TIME0662
```

```
      TM$(15,9)=0.0                                                     TIME0663
      TM$(13,10)=E2+E3                                                  TIME0664
      TM$(14,10)=E2+2.D0*E3                                             TIME0665
      TM$(15,10)=E2+E2+E3                                               TIME0666
      TM$(13,11)=0.D0                                                   TIME0667
      TM$(14,11)=0.D0                                                   TIME0668
      TM$(15,11)=E3                                                     TIME0669
      TM$(13,12)=E1+E3+E3                                               TIME0670
      TM$(14,12)=E1+E3                                                  TIME0671
      TM$(15,12)=2.D0*E1+E3                                             TIME0672
      CONST4=0.94375D0                                                  TIME0673
      TM$(1,13)=D1*CONST4                                               TIME0674
      TM$(1,14)=0.D0                                                    TIME0675
      TM$(1,15)=0.D0                                                    TIME0676
      TM$(2,13)=(D1+2.D0*D2)*CONST4                                     TIME0677
      TM$(2,14)=(2.D0*D1+D2)*CONST4                                     TIME0678
      TM$(2,15)=(D1+D2)*CONST4                                          TIME0679
      TM$(3,13)=0.D0                                                    TIME0680
      TM$(3,14)=D2*CONST4                                               TIME0681
      TM$(3,15)=0.D0                                                    TIME0682
      TM$(4,13)=(D2+D3)*CONST4                                          TIME0683
      TM$(4,14)=(D2+2.D0*D3)*CONST4                                     TIME0684
      TM$(4,15)=(2.D0*D2+D3)*CONST4                                     TIME0685
      TM$(5,13)=0.D0                                                    TIME0686
      TM$(5,14)=0.D0                                                    TIME0687
      TM$(5,15)=D3*CONST4                                               TIME0688
      TM$(6,13)=(D1+2.D0*D3)*CONST4                                     TIME0689
      TM$(6,14)=(D1+D3)*CONST4                                          TIME0690
      TM$(6,15)=(2.D0*D1+D3)*CONST4                                     TIME0691
      TM$(7,13)=E1*CONST4                                               TIME0692
      TM$(7,14)=0.D0                                                    TIME0693
      TM$(7,15)=0.D0                                                    TIME0694
      TM$(8,13)=(E1+2.D0*E2)*CONST4                                     TIME0695
      TM$(8,14)=(2.D0*E1+E2)*CONST4                                     TIME0696
      TM$(8,15)=(E1+E2)*CONST4                                          TIME0697
      TM$(9,13)=0.D0                                                    TIME0698
      TM$(9,14)=E2*CONST4                                               TIME0699
      TM$(9,15)=0.D0                                                    TIME0700
      TM$(10,13)=(E2+E3)*CONST4                                         TIME0701
      TM$(10,14)=(E2+2.D0*E3)*CONST4                                    TIME0702
      TM$(10,15)=(E2+E2+E3)*CONST4                                      TIME0703
      TM$(11,13)=0.D0                                                   TIME0704
      TM$(11,14)=0.D0                                                   TIME0705
      TM$(11,15)=E3*CONST4                                              TIME0706
      TM$(12,13)=(E1+2.D0*E3)*CONST4                                    TIME0707
      TM$(12,14)=(E1+E3)*CONST4                                         TIME0708
      TM$(12,15)=(2.D0*E1+E3)*CONST4                                    TIME0709
      CONST2=3.2981D0                                                   TIME0710
```

```
      TM$(1,16)=1.D0*CONST2
      TM$(2,17)=1.D0*CONST2
      TM$(3,18)=1.D0*CONST2
      TM$(4,19)=1.D0*CONST2
      TM$(5,20)=1.D0*CONST2
      TM$(6,21)=1.D0*CONST2

ALPHA1=ALPHA/VISCOSITY

      ALPHA1=9.4454D-05
      CONST1=ALPHA1
      TM$(16,16)=TM$(1,1)*CONST1
      TM$(16,17)=TM$(1,2)*CONST1
      TM$(16,18)=TM$(1,3)*CONST1
      TM$(16,19)=TM$(1,4)*CONST1
      TM$(16,20)=TM$(1,5)*CONST1
      TM$(16,21)=TM$(1,6)*CONST1
      TM$(17,16)=TM$(2,1)*CONST1
      TM$(17,17)=TM$(2,2)*CONST1
      TM$(17,18)=TM$(2,3)*CONST1
      TM$(17,19)=TM$(2,4)*CONST1
      TM$(17,20)=TM$(2,5)*CONST1
      TM$(17,21)=TM$(2,6)*CONST1
      TM$(18,16)=TM$(3,1)*CONST1
      TM$(18,17)=TM$(3,2)*CONST1
      TM$(18,18)=TM$(3,3)*CONST1
      TM$(18,19)=TM$(3,4)*CONST1
      TM$(18,20)=TM$(3,5)*CONST1
      TM$(18,21)=TM$(3,6)*CONST1
      TM$(19,16)=TM$(4,1)*CONST1
      TM$(19,17)=TM$(4,2)*CONST1
      TM$(19,18)=TM$(4,3)*CONST1
      TM$(19,19)=TM$(4,4)*CONST1
      TM$(19,20)=TM$(4,5)*CONST1
      TM$(19,21)=TM$(4,6)*CONST1
      TM$(20,16)=TM$(5,1)*CONST1
      TM$(20,17)=TM$(5,2)*CONST1
      TM$(20,18)=TM$(5,3)*CONST1
      TM$(20,19)=TM$(5,4)*CONST1
      TM$(20,20)=TM$(5,5)*CONST1
      TM$(20,21)=TM$(5,6)*CONST1
      TM$(21,16)=TM$(6,1)*CONST1
      TM$(21,17)=TM$(6,2)*CONST1
      TM$(21,18)=TM$(6,3)*CONST1
      TM$(21,19)=TM$(6,4)*CONST1
      TM$(21,20)=TM$(6,5)*CONST1
      TM$(21,21)=TM$(6,6)*CONST1
      IF(T.GT.0.D0) GO TO 185
```

```
      CONST3=-DEL/180.D0                                                TIME0759
      CD$(1,1)=6.D0*CONST3                                              TIME0760
      CD$(1,2)=0.D0                                                     TIME0761
      CD$(1,3)=-CONST3                                                  TIME0762
      CD$(1,4)=-4.D0*CONST3                                             TIME0763
      CD$(1,5)=-CONST3                                                  TIME0764
      CD$(1,6)=0.D0                                                     TIME0765
      CD$(2,1)=0.D0                                                     TIME0766
      CD$(2,2)=32.D0*CONST3                                             TIME0767
      CD$(2,3)=0.D0                                                     TIME0768
      CD$(2,4)=16.D0*CONST3                                             TIME0769
      CD$(2,5)=-4.D0*CONST3                                             TIME0770
      CD$(2,6)=16.D0*CONST3                                             TIME0771
      CD$(3,1)=-CONST3                                                  TIME0772
      CD$(3,2)=0.D0                                                     TIME0773
      CD$(3,3)=6.D0*CONST3                                              TIME0774
      CD$(3,4)=0.D0                                                     TIME0775
      CD$(3,5)=-CONST3                                                  TIME0776
      CD$(3,6)=-4.D0*CONST3                                             TIME0777
      CD$(4,1)=-4.D0*CONST3                                             TIME0778
      CD$(4,2)=16.D0*CONST3                                             TIME0779
      CD$(4,3)=0.D0                                                     TIME0780
      CD$(4,4)=32.D0*CONST3                                             TIME0781
      CD$(4,5)=0.D0                                                     TIME0782
      CD$(4,6)=16.D0*CONST3                                             TIME0783
      CD$(5,1)=-CONST3                                                  TIME0784
      CD$(5,2)=-4.D0*CONST3                                             TIME0785
      CD$(5,3)=-CONST3                                                  TIME0786
      CD$(5,4)=0.D0                                                     TIME0787
      CD$(5,5)=6.D0*CONST3                                              TIME0788
      CD$(5,6)=0.D0                                                     TIME0789
      CD$(6,1)=0.D0                                                     TIME0790
      CD$(6,2)=16.D0*CONST3                                             TIME0791
      CD$(6,3)=-4.D0*CONST3                                             TIME0792
      CD$(6,4)=16.D0*CONST3                                             TIME0793
      CD$(6,5)=0.D0                                                     TIME0794
      CD$(6,6)=32.D0*CONST3                                             TIME0795
      CD$(7,7)=CD$(1,1)                                                 TIME0796
      CD$(7,8)=CD$(1,2)                                                 TIME0797
      CD$(7,9)=CD$(1,3)                                                 TIME0798
      CD$(7,10)=CD$(1,4)                                                TIME0799
      CD$(7,11)=CD$(1,5)                                                TIME0800
      CD$(7,12)=CD$(1,6)                                                TIME0801
      CD$(8,7)=CD$(2,1)                                                 TIME0802
      CD$(8,8)=CD$(2,2)                                                 TIME0803
      CD$(8,9)=CD$(2,3)                                                 TIME0804
      CD$(8,10)=CD$(2,4)                                                TIME0805
      CD$(8,11)=CD$(2,5)                                                TIME0806
```

```
CD$(8,12)=CD$(2,6)                                            TIME0807
CD$(9,7)=CD$(3,1)                                             TIME0808
CD$(9,8)=CD$(3,2)                                             TIME0809
CD$(9,9)=CD$(3,3)                                             TIME0810
CD$(9,10)=CD$(3,4)                                            TIME0811
CD$(9,11)=CD$(3,5)                                            TIME0812
CD$(9,12)=CD$(3,6)                                            TIME0813
CD$(10,7)=CD$(4,1)                                            TIME0814
CD$(10,8)=CD$(4,2)                                            TIME0815
CD$(10,9)=CD$(4,3)                                            TIME0816
CD$(10,10)=CD$(4,4)                                           TIME0817
CD$(10,11)=CD$(4,5)                                           TIME0818
CD$(10,12)=CD$(4,6)                                           TIME0819
CD$(11,7)=CD$(5,1)                                            TIME0820
CD$(11,8)=CD$(5,2)                                            TIME0821
CD$(11,9)=CD$(5,3)                                            TIME0822
CD$(11,10)=CD$(5,4)                                           TIME0823
CD$(11,11)=CD$(5,5)                                           TIME0824
CD$(11,12)=CD$(5,6)                                           TIME0825
CD$(12,7)=CD$(6,1)                                            TIME0826
CD$(12,8)=CD$(6,2)                                            TIME0827
CD$(12,9)=CD$(6,3)                                            TIME0828
CD$(12,10)=CD$(6,4)                                           TIME0829
CD$(12,11)=CD$(6,5)                                           TIME0830
CD$(12,12)=CD$(6,6)                                           TIME0831
CD$(16,16)=CD$(1,1)                                           TIME0832
CD$(16,17)=CD$(1,2)                                           TIME0833
CD$(16,18)=CD$(1,3)                                           TIME0834
CD$(16,19)=CD$(1,4)                                           TIME0835
CD$(16,20)=CD$(1,5)                                           TIME0836
CD$(16,21)=CD$(1,6)                                           TIME0837
CD$(17,16)=CD$(2,1)                                           TIME0838
CD$(17,17)=CD$(2,2)                                           TIME0839
CD$(17,18)=CD$(2,3)                                           TIME0840
CD$(17,19)=CD$(2,4)                                           TIME0841
CD$(17,20)=CD$(2,5)                                           TIME0842
CD$(17,21)=CD$(2,6)                                           TIME0843
CD$(18,16)=CD$(3,1)                                           TIME0844
CD$(18,17)=CD$(3,2)                                           TIME0845
CD$(18,18)=CD$(3,3)                                           TIME0846
CD$(18,19)=CD$(3,4)                                           TIME0847
CD$(18,20)=CD$(3,5)                                           TIME0848
CD$(18,21)=CD$(3,6)                                           TIME0849
CD$(19,16)=CD$(4,1)                                           TIME0850
CD$(19,17)=CD$(4,2)                                           TIME0851
CD$(19,18)=CD$(4,3)                                           TIME0852
CD$(19,19)=CD$(4,4)                                           TIME0853
CD$(19,20)=CD$(4,5)                                           TIME0854
```

```
      CD$(19,21)=CD$(4,6)
      CD$(20,16)=CD$(5,1)
      CD$(20,17)=CD$(5,2)
      CD$(20,18)=CD$(5,3)
      CD$(20,19)=CD$(5,4)
      CD$(20,20)=CD$(5,5)
      CD$(20,21)=CD$(5,6)
      CD$(21,16)=CD$(6,1)
      CD$(21,17)=CD$(6,2)
      CD$(21,18)=CD$(6,3)
      CD$(21,19)=CD$(6,4)
      CD$(21,20)=CD$(6,5)
      CD$(21,21)=CD$(6,6)
  185 CONTINUE
      N(1)=N1
      N(2)=N2
      N(3)=N3
      N(4)=N4
      N(5)=N5
      N(6)=N6
      N(7)=N7
      N(8)=N8
      N(9)=N9
      N(10)=N10
      N(11)=N11
      N(12)=N12
      N(13)=N13
      N(14)=N14
      N(15)=N15
      N(16)=N16
      N(17)=N17
      N(18)=N18
      N(19)=N19
      N(20)=N20
      N(21)=N21
      DO 200 I$=1,21
      I=N(I$)
      DO 200 J$=1,21
      J=N(J$)
      TM(I,J)=TM(I,J)+TM$(I$,J$)
  200 CONTINUE
      IF(T.GT.0.D0) GO TO 300
      DO 210 I$=1,21
      I=N(I$)
      DO 210 J$=1,21
      J=N(J$)
      CD(I,J)=CD(I,J)+CD$(I$,J$)
  210 CONTINUE
```

```
  300 CONTINUE
      IF(NNQXY.EQ.0) GO TO 310
      DO 310 I=1,NNQXY
      RHS(NQS(I))=RHS(NQS(I))+Q(NQS(I))+65.962D0
      RHS(NQS(I)+NN)=RHS(NQS(I)+NN)+Q(NQS(I)+NN)
  310 CONTINUE
      IF(NNQZC.EQ.0) GO TO 312
      DO 312 I=1,NNQZC
      RHS(NQS(NNQXY+I)+2*NN)=RHS(NQS(NNQXY+I)+2*NN)+Q(NQS(NNQXY+I)+2*NN)
  312 CONTINUE
      IF(NNQZ.EQ.0) GO TO 311
      DO 311 I=1,NNQZ
      RHS(NQS(NNQXY+NNQZC+I)+MM)=RHS(NQS(NNQXY+NNQZC+I)+MM)+
     1Q(NQS(NNQXY+NNQZC+I)+MM)
  311 CONTINUE

MODIFICATION OF RHS FOR TM AND CD BOUNDARY CONDITIONS

      DO 315 I=1,MMM
      DO 315 J=1,NTOTVP
      JX=NVIS(J)
      RHS(I)=RHS(I)-TM(I,JX)*X(JX)
      TM(I,JX)=0.D0
      TM(JX,I)=0.D0
  315 CONTINUE
      IF(T.GT.0.D0) GO TO 321
      DO 320 I=1,NTOTVP
      K=NVIS(I)
      Y(1,K)=X(K)
      DO 316 J=1,MMM
      CD(J,K)=0.D0
  316 CD(K,J)=0.D0
      CD(K,K)=1.D0
      RHS(K)=0.D0
  320 CONTINUE
      NY=117
      NL=0
      M=70
      JSKF=0
      MAXDER=6
      IPRT=1
      H=1.D-08
      HMIN=1.D-12
      HMAX=5.D-02
      RMSEPS=1.D-03
      IF(T-0.D0) 323,323,321
  323 TEND=1.D-08
      GO TO 325
```

```
 321 TEND=T                                                              TIME0951
 325 CONTINUE                                                            TIME0952
     CALL SDESOL(Y,YL,T,TEND,NY,NL,M,JSKF,MAXDER,IPRT,H,HMIN,HMAX,       TIME0953
    1RMSEPS,W)                                                           TIME0954
     IF(T.GT.5.D-02) GO TO 324                                           TIME0955
     DO 322 J=1,MMM                                                      TIME0956
     TDIFF=DABS(T1(J)-Y(1,J))                                            TIME0957
     T1(J)=Y(1,J)                                                        TIME0958
     EPSLN=1.D-06                                                        TIME0959
     IF(TDIFF-EPSLN) 322,177,177                                         TIME0960
 322 CONTINUE                                                            TIME0961
 324 CONTINUE                                                            TIME0962
     WRITE(NWRITE,2000)                                                  TIME0963
     DO 360 I=1,MMM                                                      TIME0964
     WRITE(NWRITE,2005) I,Y(1,I)                                         TIME0965
 360 CONTINUE                                                            TIME0966
     WRITE(NWRITE,2021)                                                  TIME0967
     WRITE(NWRITE,2025)                                                  TIME0968
     WRITE(NWRITE,2030)                                                  TIME0969
     WRITE(NWRITE,2035)                                                  TIME0970
     WRITE(NWRITE,2040)                                                  TIME0971
     WRITE(NWRITE,2045)                                                  TIME0972
     WRITE(NWRITE,2050)                                                  TIME0973
     WRITE(NWRITE,2055)                                                  TIME0974
 500 FORMAT(I10)                                                         TIME0975
 600 FORMAT(1H1,18X,'TIME-DEPENDENT FLUID MECHANICS PROBLEM',////)
1005 FORMAT(3I10)                                                        TIME0976
1006 FORMAT(6X,A4,I10,2F10.0)                                            TIME0977
1010 FORMAT(7I10)                                                        TIME0978
1015 FORMAT(6X,A4,I10,F10.0)                                             TIME0979
1016 FORMAT(6X,A4,I10)                                                   TIME0980
1020 FORMAT(I10,2F10.0)                                                  TIME0981
1025 FORMAT(6X,A4,I10,F10.0)                                             TIME0982
1030 FORMAT(6X,A4,2I10,2F10.0)                                           TIME0983
1034 FORMAT(5X,'NNQZC=',I3,//)                                           TIME0984
1035 FORMAT(5X,'NO. OF NODES=',I3,//,5X,'NO. OF ELEMENTS=',I3,//,        TIME0985
    15X,'NO. OF CORNER NODES=',I3,///)                                   TIME0986
1036 FORMAT(5X,'NNVELS=',I3,//)                                          TIME0987
1037 FORMAT(5X,'NNQXY=',I3,//)                                           TIME0988
1038 FORMAT(5X,'NNPS=',I3,//)                                            TIME0989
1039 FORMAT(5X,'NNTS=',I3,//)                                            TIME0990
1040 FORMAT(5X,'NNQZ=',I3,///)                                           TIME0991
1041 FORMAT(5X,'SUMMARY OF NODAL COORDINATES',//,                        TIME0992
    17X,'I',12X'X(I)',13X,'Y(I)',//)                                     TIME0993
1045 FORMAT(5X,I3,2(7X,F10.3))                                           TIME0994
1050 FORMAT(5X,'LISTING OF SYSTEM TOPOLOGY'//,5X                         TIME0995
    1'ELEMENT NUMBER',20X,'NODE NUMBERS',//)                             TIME0996
1055 FORMAT(5X,I3,10X,6(5X,I3))                                          TIME0997
```

```
1060 FORMAT(//,7X,'NODES WHERE VELOCITIES ARE SPECIFIED',            TIME0998
    1//,8X,'I',5X,'NODE',5X,'U VELOCITY',5X,'V VELOCITY',//)         TIME0999
1065 FORMAT(2X,2(4X,I3),3X,F12.3,3X,F12.3)                           TIME1000
1070 FORMAT(//,5X,'NODES WHERE QX AND QY ARE SPECIFIED',             TIME1001
    1      //,2X,'     I     NODE',1X,'          QX',10X,'    QY',//) TIME1002
1075 FORMAT(5X,I3,2(10X,F12.3))                                      TIME1003
1080 FORMAT(//,5X,'NODES WHERE PRESSURE IS SPECIFIED',               TIME1004
    1             //,5X,'I',5X,'NODE',15X,'PRESSURE',//)             TIME1005
1081 FORMAT(//,5X,'NODES WHERE TEMPERATURE IS SPECIFIED',            TIME1006
    1             //,5X,'I',5X,'NODE',15X,'TEMPERATURE',//)          TIME1007
1082 FORMAT(//,5X,'NODES WHERE HEAT FLUX QZ IS SPECIFIED',           TIME1008
    1             //,5X,'I',5X,'NODE',15X,'HEAT FLUX',//)            TIME1009
1083 FORMAT(//,5X,'NODES WHERE QZC IS SPECIFIED',                    TIME1010
    1             //,5X,'I',5X,'NODE',23X,'QZC',//)                  TIME1011
1085 FORMAT(7X,I3,3X,I3,10X,F12.3)                                   TIME1012
1095 FORMAT(5X,3(3X,I3),2(5X,F12.3))                                 TIME1013
2000 FORMAT(///,5X,'NODE NO.',6X,'NODE VARIABLE',//)                 TIME1014
2005 FORMAT(9X,I3,5X,D17.10,/)                                       TIME1015
2015 FORMAT(1H1,5X,'THIS IS A 3-D AXISYMMETRIC PROBLEM',//)          TIME1016
2020 FORMAT(1H1,5X,'THIS IS A 2-D NON-LINEAR PROGRAM',//)            TIME1017
2021 FORMAT(//////////)                                              TIME1018
2025 FORMAT(///,10X,'THE KINEMATIC VISCOSITY OF FLUID 50-HB-3520 AT 20 TIME1019
    1DEGREES C. = 10.90 SQ.CM/SEC',//)                               TIME1020
2030 FORMAT(///,10X,'THE DENSITY OF 50-HB-3520 AT 20 DEGREES C. = 1.059TIME1021
    16 GM/CC',//)                                                    TIME1022
2035 FORMAT(///,10X,'THE COEFF. OF THERMAL EXPANSION OF 50-HB-3520 AT 2TIME1023
    10 DEGREES C. = 0.002278/DEGREE C.',//)                          TIME1024
2040 FORMAT(///,10X,'THE THERMAL DIFFUSIVITY OF 50-HB-3520 AT 20 DEGREETIME1025
    1S C. = 0.00103 SQ.CM/SEC',//)                                   TIME1026
2045 FORMAT(///,10X,'THE GRASHOF NUMBER (GR(L)) =(G*B*L**3*(TH-TC))/V**TIME1027
    12 = 946.4',//)                                                  TIME1028
2050 FORMAT(///,10X,'THE U VELOCITY FORCING FUNCTION,G*B*T(INITIAL), = TIME1029
    165.962 CM/SQ.SEC',//)                                           TIME1030
2055 FORMAT(///,10X,'THE SPECIFIED WALL PRESSURES ARE NORMALIZED TO ONETIME1031
    1 (1) ATMOSPHERE',//)                                            TIME1032
     STOP                                                            TIME1033
     END                                                             TIME1034


     SUBROUTINE SDESOL (Y,YL,T,TEND,NY,NL,M,JSKF,MAXDER,IPRT,H,HMIN,  SDE   10
    1HMAX,RMSEPS,W)                                                  SDE   20
C                                                                    SDE   30
C    ----------------------------------------------------------------SDE   40
C                                                                    SDE   50
C    SUBROUTINE SDESOL IS A DRIVER ROUTINE FOR SUBROUTINE LDASUB.    SDE   60
C    ITS PURPOSE IS TO SET UP THE NECESSARY REFERENCES TO A LARGE    SDE   70
```

```
C     BLOCK OF AUXILLARY STORAGE, AND  OBTAIN INITIAL VALUES OF
C     DERIVATIVES.
C     THE CALLING SEQUENCE FOR SDESOL IS
C
C CALL SDESOL(Y,YL,T,TEND,NY,NL,M,JSKF,MAXDER,IPRT,H,HMIN,HMAX,RMSEPS,W)
C
C     WHERE THE PARAMETERS ARE DEFINED AS FOLLOWS.
C
C        Y        - ARRAY DIMENSIONED (7,NY).  THIS ARRAY CONTAINS THE
C                   DEPENDENT VARIABLES AND THEIR SCALED DERIVATIVES.
C                   Y(J+1,I) CONTAINS THE J-TH DERIVATIVE OF THE I-TH VAR
C                   IABLE TIMES H**J/J-FACTORIAL, WHERE H IS THE CURRENT
C                   STEPSIZE.  ON FIRST ENTRY THE CALLER SUPPLIES THE
C                   INITIAL VALUES OF EACH VARIABLE IN Y(1,I).  ON SUB-
C                   SEQUENT ENTRIES IT IS ASSUMED THE ARRAY HAS NOT
C                   BEEN CHANGED.  TO INTERPOLATE TO NON-MESH POINTS,
C                   THESE VALUES CAN BE USED AS FOLLOWS.  IF H IS THE
C                   CURRENT STEPSIZE AND VALUES AT TIME  T+E  ARE
C                   NEEDED, LET  S = E/H  AND THEN
C
C                                                   JS
C                         I-TH VARIABLE AT  T+E IS  SUM Y(J+1,I)*S**J
C                                                   J=0
C
C                   THE VALUE OF JS IS OBTAINED IN THE CALLING PROGRAM
C                   BY  JS = IABS(JSKF/10)
C        YL       - ARRAY OF NL VARIABLES WHICH APPEAR LINEARLY.
C        T        - CURRENT VALUE OF THE INDEPENDENT VARIABLE (TIME)
C        TEND     - END TIME
C        NY       - NUMBER OF DIFFERENTIAL EQUATIONS AND NONLINEAR
C                   VARIABLES.
C        NL       - NUMBER OF LINEAR VARIABLES
C        M        - NUMBER OF VARIABLES INCLUDED IN THE ERROR TEST
C        JSKF     - AN INDICATOR USED BOTH ON INPUT AND OUTPUT
C                   ON INPUT, JSKF = -1 INDICATES A RESTART CALL TO
C                   SDESOL.  JSKF = 0 INDICATES AN INITIAL CALL TO
C                   SDESOL.  JSKF > 0 INDICATES A CONTINUATION OF THE
C                   PREVIOUS CALL TO SDESOL.  JSKF < -1 MAY HAVE RESULTED
C                   FROM THE USER NEGLECTING TO TEST FOR ERROR RETURNS
C                   FROM SDESOL.  BECAUSE OF THIS POSSIBILITY, JSKF < -1
C                   RESULTS IN TERMINATION OF THE RUN WITH THE
C                   APPROPRIATE COMMENT.
C                   ON OUTPUT, JSKF CONSISTS OF TWO DIGITS AND SIGN,
C                   + OR - QP.  Q IS THE ORDER OF THE FORMULA CURRENTLY
C                   BEING USED.  P INDICATES THE TYPE OF RETURN, AS
C                   FOLLOWS.
C                   JSKF > 0, P = 1 IS THE NORMAL RETURN
C                   JSKF < 0 IS AN ERROR RETURN, WITH THE FOLLOWING
```

```
C              MEANINGS.                                             SDE  560
C                   P = 1  ERROR TEST FAILURE FOR H > HMIN           SDE  570
C                   P = 3  CORRECTOR FAILED TO CONVERGE FOR H > HMINSDE  580
C                   P = 4  CORRECTOR FAILED TO CONVERGE FOR FIRST    SDE  590
C                          ORDER METHOD                              SDE  600
C                   P = 5  ERROR RETURN FROM SUBROUTINE NUITSL       SDE  610
C                   P = 6  ERROR RETURN FROM SUBROUTINE DERVAL       SDE  620
C     MAXDER   - MAXIMUM ORDER DERIVATIVE THAT SHOULD BE USED IN     SDE  630
C                METHOD.  IT MUST BE NO GREATER THAN SIX.            SDE  640
C     IPRT     - INTERNAL PRINT CONTROL INDICATOR FOR LCASUB.        SDE  650
C                     IPRT = 0  NO PRINT                             SDE  660
C                     IPRT > 0  PRINT COUNTERS, STEPSIZE, CURRENT TIMESDE  670
C                               AND VALUES OF DEPENDENT VARIABLES AT  SDE  680
C                               EACH STEP.                           SDE  690
C     H        - CURRENT STEPSIZE.  AN INITIAL VALUE MUST BE SUPPLIED SDE  700
C                BUT NEED NOT BE THE ONE WHICH MUST BE USED, SINCE THESDE  710
C                SUBROUTINE WILL CHOSE A SMALLER ONE IF NECCESSARY TO SDE  720
C                KEEP THE ERROR PER STEP SMALLER THAN THE SPECIFIED   SDE  730
C                VALUE.  IT IS BETTER TO UNDERESTIMATE THE INITIAL    SDE  740
C                STEPSIZE THAN TO OVERESTIMATE IT.  THE STEPSIZE IS   SDE  750
C                NORMALLY NOT CHANGED BY THE USER.                    SDE  760
C     HMIN     - MINIMUM STEPSIZE ALLOWED                             SDE  770
C     HMAX     - MAXIMUM STEPSIZE ALLOWED                             SDE  780
C     RMSEPS   - THE ERROR TEST CONSTANT.  THE ROOT-MEAN-SQUARE OF    SDE  790
C                THE SINGLE STEP ERROR ESTIMATES,  ER(I), DIVIDED BY  SDE  800
C                YMAX(I) = (MAXIMUM TO CURRENT TIME OF Y(I) ) MUST BE SDE  810
C                LESS THAN EPS.  THE STEPSIZE AND/OR THE ORDER        SDE  820
C                ARE VARIED TO ACHIEVE THIS.                          SDE  830
C     W        - SCRATCH STORAGE ARRAY.  MUST BE AT LEAST 13*NY + 5*NLSDE  840
C                LOCATIONS, PLUS THOSE REQUIRED FOR STORAGE OF THE    SDE  850
C                MATRIX  PW (SEE DESCRIPTION OF SUBROUTINE JACMAT).   SDE  860
C                THE STORAGE OF PW WILL NORMALLY REQUIRE NO MORE THAN SDE  870
C                N**2 + 2*N LOCATIONS, AND IF COMPACT STORAGE TECH-   SDE  880
C                NIQUES ARE USED, CAN BE MUCH FEWER.                  SDE  890
C                                                                     SDE  900
C     ----------------------------------------------------------------SDE  910
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION Y(7,1), YL(1), W(1)
      IF (JSKF.GT.0) GO TO 120
      IF (JSKF.LT.-1) GO TO 140
      N = NY+NL
      IF (JSKF.LT.0) GO TO 110
C                                                                     SDE  970
C     IF THIS IS THE FIRST ENTRY, OBTAIN VALUES OF THE DERIVATIVES.   SDE  980
      CALL DERVAL (Y,YL,T,N,NY,W,KRETR)
      IF (KRETR.NE.0) GO TO 130
C                                                                     SDE 1010
C     NOW SET UP STORAGE BLOCKS IN THE W ARRAY.  THIS NEEDS TO BE DONE SDE 1020
```

```
C     ONLY INITIALLY AND ON RESTARTS.                                   SDE 1030
C                                                                       SDE 1040
C         THE ARRAY   SAVE    STARTS AT LOCATION  1     IN THE W ARRAY  SDE 1050
C         THE ARRAY   YLSV    STARTS AT LOCATION  NSVL  IN THE W ARRAY  SDE 1060
C         THE ARRAY   YMAX    STARTS AT LOCATION  NYMAX IN THE W ARRAY  SDE 1070
C         THE ARRAY   ER      STARTS AT LOCATION  NER   IN THE W ARRAY  SDE 1080
C         THE ARRAY   ESV     STARTS AT LOCATION  NESV  IN THE W ARRAY  SDE 1090
C         THE ARRAY   F1      STARTS AT LOCATION  NF1   IN THE W ARRAY  SDE 1100
C         THE ARRAY   DY      STARTS AT LOCATION  NDY   IN THE W ARRAY  SDE 1110
C         THE MATRIX  PW      STARTS AT LOCATION  NPW   IN THE W ARRAY  SDE 1120
C                                                                       SDE 1130
  110 NSVL = 7*NY+1
      NYMAX = NSVL+NL
      NER = NYMAX+NY
      NESV = NER+NY
      NF1 = NESV+NY
      NDY = NF1+N
      NPW = NDY+N
  120 JS = JSKF
      CALL LDASUB (Y,YL,T,TEND,N,NY,M,JS,KF,MAXDER,IPRT,H,HMIN,HMAX,
     1RMSEPS,W,W(NSVL),W(NYMAX),W(NER),W(NESV),W(NF1),W(NDY),W(NPW))
C                                                                       SDE 1240
C     CODE JSKF ON RETURN FROM LDASUB                                   SDE 1250
C                                                                       SDE 1260
      JSKF = ISIGN(JS*10+IABS(KF),KF)
      RETURN
  130 JSKF = -6
      RETURN
  140 PRINT 1, JSKF
      STOP
C                                                                       SDE 1330
C                                                                       SDE 1340
    1 FORMAT ('0IT IS AN ERROR TO ENTER SDESOL WITH JSKF = ',I10//
     1 '   RUN HAS BEEN TERMINATED.')
      END
      SUBROUTINE LDASUB (Y,YL,T,TEND,N,NY,M,JSTART,KFLAG,MAXOR,IPRT,H,
     1HMIN,HMAX,RMSEPS,SAVE,YLSV,YMAX,ER,ESV,F1,DY,PW)
C                                                                       LDA   30
C     SUBROUTINE LDASUB IS A MODIFICATION OF SUBROUTINE DFASUB          LDA   40
C     WHICH IS DUE TO R. L. BROWN AND C. W. GEAR.  DFASUB IS DOCUMENTED LDA   50
C     IN THE REPORT                                                     LDA   60
C         DOCUMENTATION FOR DFASUB--                                    LDA   70
C         BY R. L. BROWN AND C. W. GEAR                                 LDA   80
C         REPORT UIUCDCS-R-73-575, JULY 1973                            LDA   90
C         UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN                    LDA  100
C         URBANA, ILLINOIS  61801                                       LDA  110
C     THIS REPORT IS AVAILABLE FROM THE NATIONAL TECHNICAL INFORMATION  LDA  120
C     SERVICE OF THE U. S. DEPARTMENT OF COMMERCE UNDER ACCESSION NUMBERLDA  130
```

```
C       CCO-1469-225.
C
C       THE MODIFICATION HERE IS DOCUMENTED IN THE REPORT
C       A PROGRAM FOR THE NUMERICAL SOLUTION OF LARGE SPARSE SYSTEMS OF
C       ALGEBRAIC AND IMPLICITLY DEFINED STIFF DIFFERENTIAL EQUATIONS
C          BY RICHARD FRANKE
C          REPORT NPS53FE76051, MAY 1976
C          NAVAL POSTGRADUATE SCHOOL
C          MONTEREY, CALIFORNIA  93940
C
C       -----------------------------------------------------------------
C
C       THE CALLING SEQUENCE FOR LDASUB IS
C
C       CALL LDASUB(Y,YL,T,TEND,N,NY,M,JSTART,KFLAG,MAXOR,IPRT,H,HMIN,
C       HMAX,RMSEPS,SAVE,YLSV,YMAX,ER,ESV,F1,DY,PW)
C
C       WHERE THE PARAMETERS ARE DEFINED AS FOLLOWS.
C          Y         - ARRAY DIMENSIONED (7,NY).  THIS ARRAY CONTAINS THE
C                      DEPENDENT VARIABLES AND THEIR SCALED DERIVATIVES.
C                      Y(J+1,I) CONTAINS THE J-TH DERIVATIVE OF THE I-TH VAR
C                      IABLE TIMES H**J/J-FACTORIAL, WHERE H IS THE CURRENT
C                      STEPSIZE.  ON FIRST ENTRY THE CALLER SUPPLIES THE
C                      INITIAL VALUES OF EACH VARIABLE IN Y(1,I) AND AN
C                      ESTIMATE OF THE INITIAL VALUES OF THE DERIVATIVES
C                      IN Y(2,I).  ON SUBSEQUENT ENTRIES IT IS ASSUMED THAT
C                      THE ARRAY HAS NOT BEEN CHANGED.  TO INTERPOLATE TO
C                      NON-MESH POINTS, THESE VALUES CAN BE USED AS FOLLOWS.
C                      IF H IS THE CURRENT STEPSIZE AND VALUES AT TIME T+E
C                      NEEDED, LET S = E/H  AND THEN
C
C                                                     NQ
C                           I-TH VARIABLE AT  T+E IS  SUM Y(J+1,I)*S**J
C                                                     J=0
C
C                      THE VALUE OF NQ IS OBTAINED IN THE CALLING PROGRAM
C                      BY  NQ = JSTART.
C
C          YL        - ARRAY OF NL = N - NY VARIABLES WHICH APPEAR LINEARLY.
C                      THE USER SUPPLIES INITIAL VALUES FOR THESE VARIABLES.
C          T         - CURRENT VALUE OF THE INDEPENDENT VARIABLE (TIME)
C          TEND      - END TIME
C          N         - TOTAL NUMBER OF VARIABLES
C          NY        - NUMBER OF DIFFERENTIAL EQUATIONS AND NONLINEAR
C                      VARIABLES.
C          M         - NUMBER OF VARIABLES INCLUDED IN THE ERROR TEST.
C                      THIS NUMBER CAN BE NO GREATER THAN NY.  IF IT IS
C                      GREATER THAN NY, NY VARIABLES ARE USED IN THE ERROR
```

```
C        TEST.                                                            LDA  620
C JSTART  - INPUT AND OUTPUT INDICATOR.                                   LDA  630
C           ON INPUT JSTART HAS THE FOLLOWING MEANINGS.                   LDA  640
C                <0  THIS INDICATES A RE-START FROM A PREVIOUS            LDA  650
C                    POINT FOLLOWING TERMINATION OF THE RUN OR            LDA  660
C                    SOLUTION OF ANOTHER PROBLEM DURING THE SAME          LDA  670
C                    RUN.  PARAMETERS IN THE CALLING SEQUENCE             LDA  680
C                    MUST HAVE BEEN PRESERVED FROM THE PREVIOUS           LDA  690
C                    USE, PARTICULARLY THE ARRAYS                         LDA  700
C                         SAVE, YLSV, ESV, AND PW.                        LDA  710
C                    THESE ARRAYS MUST BE SAVED AFTER A CALL              LDA  720
C                    TO SUBROUTINE LDASAV, WHICH ALSO SAVES               LDA  730
C                    NECESSARY PARAMETERS INTERNAL TO LDASUB.             LDA  740
C                =0  INDICATES AN INITIAL CALL TO LDASUB.  THE            LDA  750
C                    ROUTINE INITIALIZES ITSELF, SCALES THE               LDA  760
C                    DERIVATIVES IN Y(2,I) AND THEN PERFORMS THE          LDA  770
C                    INTEGRATION UNTIL T > TEND.                          LDA  780
C                >0  INDICATES THE SOLUTION IS TO BE CONTINUED.           LDA  790
C                    AFTER THE INITIAL ENTRY IT IS NEITHER                LDA  800
C                    DESIRABLE NOR NECESSARY TO RE-ENTER WITH             LDA  810
C                    JSTART = 0, SINCE THIS RE-INITIALIZES                LDA  820
C                    THE CODE, BEGINNING WITH A FIRST ORDER               LDA  830
C                    METHOD AGAIN.                                        LDA  840
C           ON OUTPUT, JSTART IS SET TO THE VALUE OF NQ, THE              LDA  850
C           ORDER OF THE FORMULA CURRENTLY BEI-& O & &                   LDA  860
C KFLAG   - THE COMPLETION CODE INDICATOR, WITH THE FOLLOWING             LDA  870
C           MEANINGS                                                      LDA  880
C                +1  THE INTEGRATION WAS SUCCESSFUL                       LDA  890
C                -1  ERROR TEST FAILURE FOR H > HMIN                      LDA  900
C                -3  CORRECTOR FAILED TO CCNVERGE FOR H > HMIN            LDA  910
C                -4  CORRECTOR FAILED TO CCNVERGE FOR FIRST               LDA  920
C                    ORDER METHOD                                         LDA  930
C                -5  ERROR RETURN FROM SUBROUTINE NUITSL                  LDA  940
C MAXOR   - MAXIMUM ORDER DERIVATIVE THAT SHOULD BE USED IN THE           LDA  950
C           METHOD.  IT MUST BE NO GREATER THAN SIX.  IF IT IS            LDA  960
C           GREATER THAN SIX, THE MAXIMUM ORDER USED WILL BE SIX.LDA  970
C IPRT    - INTERNAL PRINT CONTROL INDICATOR                              LDA  980
C                = 0  NO PRINT                                            LDA  990
C                > 0  PRINT COUNTERS, STEPSIZE, CURRENT TIME              LDA 1000
C                     AND VALUES OF DEPENDENT VARIABLES AT                LDA 1010
C                     EACH STEP.                                          LDA 1020
C H       - CURRENT STEPSIZE.  AN INITIAL VALUE MUST BE SUPPLIED LDA 1030
C           BUT NEED NOT BE THE ONE WHICH WILL BE USED, SINCE THELDA 1040
C           SUBROUTINE WILL CHOOSE A SMALLER ONE IF NECCESSARY TOLDA 1050
C           KEEP THE ERROR PER STEP SMALLER THAN THE SPECIFIED            LDA 1060
C           VALUE.  IT IS BETTER TO UNDERESTIMATE THE INITIAL             LDA 1070
C           STEPSIZE THAN TO OVERESTIMATE IT.  THE STEPSIZE IS            LDA 1080
C           NORMALLY NOT CHANGED BY THE USER.                             LDA 1090
```

```
C        HMIN      - MINIMUM STEPSIZE ALLOWED
C        HMAX      - MAXIMUM STEPSIZE ALLOWED
C        RMSEPS    - THE ERROR TEST CONSTANT.  THE ROOT-MEAN-SQUARE OF
C                    THE SINGLE STEP ERROR ESTIMATES,  ER(I), DIVIDED BY
C                    YMAX(I) = (MAXIMUM TO CURRENT TIME OF Y(I)) MUST BE
C                    LESS THAN RMSEPS.  THE STEPSIZE AND/OR ORDER ARE
C                    VARIED TO ACHIEVE THIS.
C        SAVE      - AN ARRAY OF LENGTH AT LEAST 7*NY
C        YLSV      - AN ARRAY OF LENGTH AT LEAST NL
C        YMAX      - A VECTOR OF LENGTH NY WHICH CONTAINS THE MAXIMUM
C                    OF EACH Y SEEN SO FAR.  ON THE FIRST CALL, THESE WILL
C                    BE INITIALIZED AS YMAX(I) = MAX(1,|Y(1,I)|)
C        ER        - A VECTOR OF LENGTH NY
C        ESV       - A VECTOR OF LENGTH NY
C        F1        - A VECTOR OF LENGTH N = NY + NL
C        DY        - A VECTOR OF LENGTH N = NY + NL
C        PW        - AN ARRAY IN WHICH THE  J  MATRIX COMPUTED
C                    IN SUBROUTINE JACMAT WILL BE STORED.  SIZE WHICH
C                    MUST BE ALLOWED IS DETERMINED BY THE STORAGE TECH-
C                    NIQUE USED FOR IT, BUT NORMALLY WON'T BE MORE THAN
C                    N**2 + 2*N LOCATIONS, THE LATTER 2*N BEING REQUIRED
C                    BY THE LINEAR EQUATION SOLVER.
C
C     ----------------------------------------------------------------------
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION Y(7,1), YL(1), SAVE(7,1), YMAX(1), ER(1), YLSV(1), F1(1)
     1, PERT(6,3), COF(21), ESV(1), DY(1), PW(1), SAV(1), A(29)
      EQUIVALENCE (A(8),BND), (A(9),BR), (A(10),E), (A(11),EDWN),
     1(A(12),ENQ1), (A(13),ENQ2), (A(14),ENQ3), (A(15),EPS), (A(16),EUP)
     2,(A(17),HNEW), (A(18),PEPSH), (A(19),IDOUB), (A(20),IWEVAL),
     3(A(21),K), (A(22),LCOPYL), (A(23),LCOPYY), (A(24),MAXDER),
     4(A(25),M1), (A(26),NL), (A(27),NQ), (A(28),NS), (A(29),NW)
C
C     ----------------------------------------------------------------------
C
C     THE COEFFICIENTS IN THE PERT ARRAY ARE USED FOR ERROR TESTING AND
C     CHANGING STEPSIZE AND NEED TO BE ACCURATE TO ONLY A FEW DIGITS.
C
C     ----------------------------------------------------------------------
      DATA PERT/4.D0,9.D0,16.D0,25.D0,36.D0,49.D0,9.D0,16.D0,25.D0,36.D0
     1,49.D0,64.D0,1.D0,1.D0,.25D0, 2.7889D-2,1.70569D-3,6.83929D-5/
C     ----------------------------------------------------------------------
C
C     THE ENTRIES IN THE COF ARRAY ARE THE COEFFICIENTS FOR THE STIFFLY
C     STABLE METHODS USED IN THIS PROGRAM AND ARE TO BE THE MACHINE
C     PRECISION EQUIVALENTS OF THE FOLLOWING CONSTANTS.
C
C     -1
```

```
C     -3/2 , -1/2                                                       LDA 1570
C     -11/6 , -1 , -1/6                                                 LDA 1580
C     -25/12 , -35/24 , -5/12 , -1/24                                   LDA 1590
C     -137/60 , -15/8 , -17/24 , -1/8 , -1/120                          LDA 1600
C     -147/60, -203/90 , -49/48 , -35/144 , -7/240 , -1/720             LDA 1610
C                                                                       LDA 1620
C     ------------------------------------------------------------------LDA 1630
      DATA COF/-1.D0,-1.5D0,-.5D0,-1.8333333333333333D0,-1.D0,
     1-1.6666666666666667D-1,-2.0833333333333333D0,-1.4583333333333333D0,
     2-4.1666666666666667D-1,-4.1666666666666667D-2,-2.2833333333333333D0,
     3 -1.875D0,-7.0833333333333333D-1,-1.25D-1,-8.3333333333333333D-3,
     4-2.45D0,-2.2555555555555556D0,-1.0208333333333333D0,
     5-2.4305555555555556D-1,-2.9166666666666667D-2,-1.3888888888888889D-3/
      IF (JSTART) 100,110,150
C     ------------------------------------------------------------------LDA 1680
C     IF THIS IS A RESTART ENTRY, RESTORE Y AND YL FROM THE SAVE AND    LDA 1690
C     YLSV ARRAYS, WHERE THEY WERE SAVED BY A PREVIOUS CALL TO LDASAV.  LDA 1700
C     ------------------------------------------------------------------LDA 1710
  100 CALL COPYZ (Y,SAVE,LCOPYY)
      CALL COPYZ (YL,YLSV,LCOPYL)
      GO TO 150
C     ------------------------------------------------------------------LDA 1750
C     IF THIS IS THE FIRST CALL, INITIALIZE YMAX, SCALE DERIVATIVES, ANDLDA 1760
C     INITIALIZE INDICATORS AND SET ORDER TO ONE.                       LDA 1770
C     FOR DOUBLE PRECISION, SET LCOPYL = 14*NY AND LCOPYL = 2*NL IF     LDA 1780
C     SUBROUTINE COPYZ IS IN SINGLE PRECISION.                          LDA 1790
C     ------------------------------------------------------------------LDA 1800
  110 NL = N-NY
      LCOPYY = 7*NY
      LCOPYL = NL
      M1 = MINO(M,NY)
      EPS =DSQRT(DFLOAT(M1))*RMSEPS
      MAXDER = MINO(MAXOR,6)
      IF (IPRT.LE.0) GO TO 120
      PRINT 3, N,NL,RMSEPS,TEND,H
      PRINT 4
  120 NS = 0
      NW = 0
C                                                                       LDA 1920
      DO 130 J=1,NY
      YMAX(J) =DMAX1(1.D0,DABS(Y(1,J)))
  130 Y(2,J) = Y(2,J)*H
C                                                                       LDA 1960
      NC = 1
      BR = 1.D0
      ASSIGN 190 TO IRET
C     ------------------------------------------------------------------LDA 2000
C     SET COEFFICIENTS FOR THE ORDER CURRENTLY BEING USED.              LDA 2010
```

```
C     E IS A TEST FOR ERRORS OF THE CURRENT ORDER NQ                    LDA 2020
C     EUP IS TO TEST FOR INCREASING THE ORDER, EDWN FOR DECREASING THE  LDA 2030
C     ORDER.                                                            LDA 2040
C     -----------------------------------------------------------------LDA 2050
  140 K = NQ*(NQ-1)/2
      CALL COPYZ (A(2),COF(K+1),NQ)
      K = NQ+1
      IDOUB = NQ
      ENQ1 = .5D0/NQ
      ENQ2 = .5D0/K
      ENQ3 = .5D0/(NQ+2)
      PEPSH = EPS**2
      E = PERT(NQ,1)*PEPSH
      EUP = PERT(NQ,2)*PEPSH
      EDWN = PERT(NQ,3)*PEPSH
      BND = (EPS*ENQ3)**2
      IWEVAL = 1
      GO TO IRET, (190,200,490,570)
  150 IF (H.EQ.HNEW) GO TO 190
C     -----------------------------------------------------------------LDA 2210
C     IF CALLER HAS CHANGED H, RESCALE DERIVATIVES TO REFLECT THAT HNEW LDA 2220
C     WAS USED ON THE LAST CALL.                                        LDA 2230
C     -----------------------------------------------------------------LDA 2240
      R = H/HNEW
      ASSIGN 190 TO IRET
      GO TO 610
C     -----------------------------------------------------------------LDA 2280
C     SET JSTART TO NQ, THE CURRENT ORDER OF THE METHOD, BEFORE EXIT,   LDA 2290
C     AND SAVE THE CURRENT STEPSIZE IN HNEW.                            LDA 2300
C     ---------------------------------------------------------------- LDA 2310
  160 JSTART = NQ
      HNEW = H
      RETURN
  170 NS = NS+1
      IF (IPRT.LE.0) GO TO 180
C     -----------------------------------------------------------------LDA 2370
C     PRINT DATA IF DESIRED BY USER                                     LDA 2380
C     -----------------------------------------------------------------LDA 2390
      PRINT 1, NS,NW,NQ,H,T,(Y(1,I),I=1,NY)
      IF (NL.GT.0) PRINT 2, (YL(I),I=1,NL)
  180 CONTINUE
      IF (KFLAG.LT.0) GO TO 160
      IF (T.GE.TEND) GO TO 160
C     -----------------------------------------------------------------LDA 2450
C     TAKE ANOTHER STEP IF T < TEND                                     LDA 2460
C     -----------------------------------------------------------------LDA 2470
      JSTART = 1
C     -----------------------------------------------------------------LDA 2490
```

```
C       SAVE DATA FOR TRIAL WITH A SMALLER TIMESTEP IF THIS STEP FAILS      LDA 2500
C       -------------------------------------------------------------------LDA 2510
  190   CALL COPYZ (SAVE,Y,LCOPYY)
        CALL COPYZ (YLSV,YL,LCOPYL)
        RACUM = 1.DO
        KFLAG = 1
        HOLD = H
        NCOLD = NQ
        TOLD = T
  200   T = T+H
        HINV = 1.DO/H
C       -------------------------------------------------------------------LDA 2610
C       COMPUTE PREDICTED VALUES BY EFFECTIVELY MULTIPLYING DERIVATIVE      LDA 2620
C       VECTOR BY PASCAL TRIANGLE MATRIX                                    LDA 2630
C       -------------------------------------------------------------------LDA 2640
C                                                                           LDA 2650
        DO 210 J=2,K
        J3 = K+J-1
C                                                                           LDA 2680
        DO 210 J1=J,K
        J2 = J3-J1
C                                                                           LDA 2710
        DO 210 I=1,NY
  210   Y(J2,I) = Y(J2,I)+Y(J2+1,I)
C                                                                           LDA 2740
C                                                                           LDA 2750
        DO 220 I=1,NY
  220   ER(I) = 0.DO
C                                                                           LDA 2780
C       -------------------------------------------------------------------LDA 2790
C       DO UP TO THREE CORRECTOR ITERATIONS.  CONVERGENCE IS OBTAINED WHENLDA 2800
C       CHANGES ARE LESS THAN BND WHICH IS DEPENDENT ON THE ERROR TEST      LDA 2810
C       CONSTANT.  THE SUM OF CORRECTIONS IS ACCUMULATED IN ER(I).  IT IS LDA 2820
C       EQUAL TO THE K-TH DERIVATIVE OF Y TIMES H**K/(K-FACTORIAL*A(K)),  LDA 2830
C       AND THUS IS PROPORTIONAL TO THE ACTUAL ERRORS TO THE LOWEST POWER LDA 2840
C       OF H PRESENT, WHICH IS H**K.                                        LDA 2850
C       -------------------------------------------------------------------LDA 2860
C                                                                           LDA 2870
        DO 270 L=1,3
        CALL DIFFUN (Y,YL,T,HINV,DY)
        IF (IWEVAL.LT.1) GO TO 230
C       -------------------------------------------------------------------LDA 2910
C       IF THERE HAS BEEN A CHANGE OF ORDER OR THERE HAS BEEN TROUBLE       LDA 2920
C       WITH CONVERGENCE, PW IS RE-EVALUATED PRIOR TO STARTING THE          LDA 2930
C       CORRECTOR ITERATION.  IWEVAL IS THEN SET TO -1 AS AN INDICATOR      LDA 2940
C       THAT IT HAS BEEN DONE.  NEWPW IS SET NONZERO TO INDICATE TO         LDA 2950
C       SUBROUTINE NUITSL THAT A NEW PW HAS BEEN PROVIDED.                  LDA 2960
C       -------------------------------------------------------------------LDA 2970
```

```
      CALL JACMAT (Y,YL,T,HINV,A(2),N,NY,EPS,DY,F1,PW)
      KFLAG = 1
      IWEVAL = -1
      NW = NW+1
      NEWPW = 1
  230 CALL NUITSL (PW,DY,F1,N,NY,EPS,YMAX,NEWPW,KRRET)
      IF (KRRET.NE.0) GO TO 600
      IF (NL.LE.0) GO TO 250
C                                                                       LDA 3060
      DO 240 I=1,NL
  240 YL(I) = YL(I)-F1(I+NY)
C                                                                       LDA 3090
  250 CONTINUE
      DEL = 0.D0
C                                                                       LDA 3120
      DC 260 I=1,NY
      Y(1,I) = Y(1,I)-F1(I)
      Y(2,I) = Y(2,I)+A(2)*F1(I)
      ER(I) = ER(I)+F1(I)
      DEL = DEL+(F1(I)/DMAX1(YMAX(I),DABS(Y(1,I))))**2
  260 CCNTINUE
C                                                                       LDA 3190
      IF (L.GE.2) BR =DMAX1(.9D0*BR,DEL/DEL1)
      DEL1 = DEL
      IF (DMIN1(DEL,BR*DEL*2.D0).LE.BND) GO TO 330
  270 CONTINUE
C                                                                       LDA 3240
C     ------------------------------------------------------------------LDA 3250
C     THE CORRECTIOR ITERATION FAILED TO CONVERGE IN 3 TRIES.  VARIOUS  LDA 3260
C     POSSIBILITIES ARE CHECKED FOR.  IF H IS ALREADY HMIN AND PW HAS   LDA 3270
C     ALREADY BEEN RE-EVALUATED, A NO CCNVERGENCE EXIT IS TAKEN.        LDA 3280
C     OTHERWISE THE MATRIX PW IS RE-EVALUATED AND/OR (IN THAT ORDER) THELDA 3290
C     STEP IS REDUCED TO TRY AND GET CONVERGENCE.                       LDA 3300
C     ------------------------------------------------------------------LDA 3310
      T = TOLD
      IF (IWEVAL) 280,300,290
  280 IF (H.LE.HMIN*1.00001D0) GO TO 310
  290 RACUM = RACUM*.25D0
  300 CONTINUE
      GO TO 560
  310 KFLAG = -3
C     ------------------------------------------------------------------LDA 3390
C     RESTCRE Y AND YL AFTER CONVERGENCE FAILURE                        LDA 3400
C     ------------------------------------------------------------------LDA 3410
  320 CALL COPYZ (Y,SAVE,LCOPYY)
      CALL COPYZ (YL,YLSV,LCOPYL)
      H = HOLD
      NC = NQOLD
```

```
      GO TO 170
C     ----------------------------------------------------------------LDA 3470
C     THE CORRECTOR CONVERGED, SO NOW THE ERROR TEST IS MADE.          LDA 3480
C     ----------------------------------------------------------------LDA 3490
  330 D = 0.D0
C                                                                      LDA 3510
      DO 340 I=1,M1
      YM =DMAX1(DABS(Y(1,I)),YMAX(I))
  340 D = D+(ER(I)/YM)**2
C                                                                      LDA 3550
      IWEVAL = 0
      IF (D.GT.E) GO TO 380
C     ----------------------------------------------------------------LDA 3580
C     THE ERROR TEST IS OKAY, SO THE STEP IS ACCEPTED.  IF IDOUB       LDA 3590
C     NOW BECOMES NEGATIVE, A TEST IS MADE TO SEE IF THE STEP SIZE     LDA 3600
C     CAN BE INCREASED AT THIS ORDER OR ONE HIGHER OR ONE LOWER.       LDA 3610
C     THE CHANGE IS MADE ONLY IF THE STEP CAN BE INCREASED BY AT       LDA 3620
C     LEAST 10%.  IDOUB IS SET TO NQ TO PREVENT FURTHER TESTING        LDA 3630
C     FOR A WHILE.  IF NO CHANGE IS MADE, IDOUB IS SET TO 9.           LDA 3640
C     ----------------------------------------------------------------LDA 3650
      IF (K.LT.3) GO TO 360
C                                                                      LDA 3670
      DO 350  J=3,K
C                                                                      LDA 3690
      DO 350  I=1,NY
  350 Y(J,I)  = Y(J,I)+A(J)*ER(I)
C                                                                      LDA 3720
  360 KFLAG = 1
      IDOUB = IDOUB-1
      IF (IDOUB) 410,370,510
  370 CALL COPYZ (ESV,ER,M1)
      GO TO 510
C     ----------------------------------------------------------------LDA 3780
C     THE ERROR TEST FAILED.  IF JSTART = 0, THE DERIVATIVES IN THE    LDA 3790
C     SAVE ARRAY ARE UPDATED.  TESTS ARE THEN MADE TO FIX THE STEPSIZE LDA 3800
C     AND PERHAPS REDUCE THE ORDER.  AFTER RESTORING AND SCALING THE   LDA 3810
C     Y VARIABLES, THE STEP IS RETRIED.                                LDA 3820
C     ----------------------------------------------------------------LDA 3830
  380 IF (JSTART.GT.0) GO TO 400
C                                                                      LDA 3850
      DO 390 I=1,NY
  390 SAVE(2,I) = Y(2,I)
C                                                                      LDA 3880
  400 KFLAG = KFLAG-2
      IF (H.LE.HMIN) GO TO 550
      T = TOLD
      IF (KFLAG.LE.-5) GO TO 530
  410 PR2 = (D/E)**ENQ2*1.2D0
```

```
      L = 0
      IF (NQ.LE.1) GO TO 430
      D = 0.D0
C                                                                       LDA 3970
      DO 420 J=1,M1
      YM =DMAX1(DABS(Y(1,J)),YMAX(J))
  420 D = D+(Y(K,J)/YM)**2
C                                                                       LDA 4010
      PR1 = (D/EDWN)**ENQ1*1.3D0
      IF (PR1.GE.PR2) GO TO 430
      PR2 = PR1
      L = -1
  430 IF (KFLAG.LT.0.OR.NQ.GE.MAXDER) GO TO 450
      D = 0
C                                                                       LDA 4080
      DO 440 J=1,M1
      YM =DMAX1(DABS(Y(1,J)),YMAX(J))
  440 D = D+((ER(J)-ESV(J))/YM)**2
C                                                                       LDA 4120
      PR1 = (D/EUP)**ENQ3*1.4D0
      IF (PR1.GE.PR2) GO TO 450
      PR2 = PR1
      L = 1
  450 R = 1.D0/DMAX1(PR2,1.D-5)
      IF (KFLAG.LT.0.OR.R.GE.1.1D0) GO TO 460
      IDOUB = 9
      GO TO 510
  460 NEWQ = NQ+L
      K = NEWQ+1
      IF (NEWQ.LE.NQ) GO TO 480
      R1 = A(NEWQ)/DFLOAT(NEWQ)
C                                                                       LDA 4250
      DO 470 J=1,NY
  470 Y(K,J) = ER(J)*R1
C                                                                       LDA 4280
  480 CONTINUE
C     ------------------------------------------------------------------LDA 4300
C     IF THE STEP WAS OKAY, SCALE THE Y VARIABLES IN ACCORDANCE         LDA 4310
C     WITH THE NEW VALUE OF H.  IF KFLAG < 0, HOWEVER, USE THE          LDA 4320
C     SAVED VALUES (IN SAVE AND YLSV).  IN EITHER CASE, IF THE ORDER    LDA 4330
C     HAS CHANGED IT IS NECESSARY TO FIX CERTAIN PARAMETERS BY CALLING  LDA 4340
C     THE PROGRAM SEGMENT AT STATEMENT NUMBER 140.                      LDA 4350
C     ------------------------------------------------------------------LDA 4360
      IDOUB = NQ
      IF (NEWQ.EQ.NQ) GO TO 490
      NQ = NEWQ
      ASSIGN 490 TO IRET
      GO TO 140
```

```
  490 IF (KFLAG.GT.0) GO TO 500
      RACUM = RACUM*R
      GO TO 560
  500 R =DMAX1(DMIN1(HMAX/H,R),HMIN/H)
      H = H*R
      IWEVAL = 1
      ASSIGN 510 TO IRET
      GO TO 610
C                                                                       LDA 4500
  510 DO 520 I=1,M1
  520 YMAX(I) =DMAX1(DABS(Y(1,I)),YMAX(I))
C                                                                       LDA 4530
      GO TO 170
C     -----------------------------------------------------------------LDA 4550
C     THE ERROR TEST HAS NOW FAILED THREE TIMES, SO THE DERIVATIVES ARE LDA 4560
C     IN BAD SHAPE.  RETURN TO FIRST ORDER METHOD AND TRY AGAIN.  OF    LDA 4570
C     COURSE, IF NQ = 1 ALREADY, THEN THERE IS NO HOPE AND WE EXIT WITH LDA 4580
C     KFLAG = -4.                                                       LDA 4590
C     -----------------------------------------------------------------LDA 4600
  530 IF (NQ.EQ.1) GO TO 540
      NQ = 1
      IDOUB = 1
      ASSIGN 570 TO IRET
      GO TO 140
  540 NCOLD = 1
      KFLAG = -4
      GO TO 320
  550 KFLAG = -1
      GO TO 170
C     -----------------------------------------------------------------LDA 4710
C     THIS SECTION RESTORES THE SAVED VALUES OF Y AND YL, SCALING THE   LDA 4720
C     Y DERIVATIVES AS NECESSARY, AND THEN RETURNS TO THE PREDICTOR LOOPLDA 4730
C     -----------------------------------------------------------------LDA 4740
  560 H = HOLD*RACUM
      H =DMAX1(HMIN,DMIN1(H,HMAX))
  570 RACUM = H/HOLD
      R1 = 1.D0
C                                                                       LDA 4790
      DO 580 J=2,K
      R1 = R1*RACUM
C                                                                       LDA 4820
      DO 580 I=1,NY
  580 Y(J,I) = SAVE(J,I)*R1
C                                                                       LDA 4850
C                                                                       LDA 4860
      DO 590 I=1,NY
  590 Y(1,I) = SAVE(1,I)
C                                                                       LDA 4890
```

```
      CALL COPYZ (YL,YLSV,LCOPYL)
      IWEVAL = 1
      GO TO 200
  600 KFLAG = -5
      GO TO 160
C     ----------------------------------------------------------------LDA 4950
C     THIS SECTION SCALES THE Y DERIVATIVES BY R**J                    LDA 4960
C     ---------------------------------------------------------------- LDA 4970
  610 R1 = 1.D0
C                                                                      LDA 4990
      DO 620 J=2,K
      R1 = R1*R
C                                                                      LDA 5020
      DO 620 I=1,NY
  620 Y(J,I) = Y(J,I)*R1
      GO TO IRET, (190,510)
C                                                                      LDA 5050
C     ----------------------------------------------------------------LDA 5070
C     THIS SECTION ALLOWS FOR RESTARTS AFTER SOLVING ANOTHER PROBLEM, ORLDA 5080
C     HAVING TERMINATED THE CURRENT COMPUTER RUN.  SUBROUTINE LDASAV   LDA 5090
C     SAVES THE NECESSARY VALUES WHICH ARE INTERNAL TO LDASUB.  FOR    LDA 5100
C     DOUBLE PRECISION, WITH COPYZ IN SINGLE PRECISION, THE NUMBER OF  LDA 5110
C     LOCATIONS TO BE SAVED AND RESTORED, LCOPYS AND LCOPYR, MUST BE   LDA 5120
C     SET TO 58.                                                       LDA 5130
C     IT IS ASSUMED THAT IN ADDITION TO THE VARIABLES IN THE ARRAY A   LDA 5140
C     SAVED BY CALLING LDASAV, THE USER ALSO SAVES THE ARRAYS SAVE,    LDA 5150
C     YLSV, YMAX, ESV, AND PW.                                         LDA 5160
C                                                                      LDA 5170
C     TO RESTART THE USER FIRST CALLS LDARST TO RESTORE THE VALUES SAVEDLDA 5180
C     BY LDASAV, THEN RE-ENTERS LDASUB WITH JSTART < 0, AND WITH THE   LDA 5190
C     OTHER PARAMETERS THE SAME AS RETURNED FROM THE LAST ENTRY TO     LDA 5200
C     LDASUB, PARTICULARLY THOSE ARRAYS MENTIONED ABOVE.               LDA 5210
C     ----------------------------------------------------------------LDA 5220
      ENTRY LDASAV(SAV)
      LCOPYS = 29
      CALL COPYZ (SAV,A,LCOPYS)
      CALL COPYZ (SAVE,Y,LCOPYY)
      CALL COPYZ (YLSV,YL,LCOPYL)
      RETURN
C                                                                      LDA 5290
      ENTRY LDARST(SAV)
      LCOPYR = 29
      CALL COPYZ (A,SAV,LCOPYR)
      RETURN
C                                                                      LDA 5340
C     ----------------------------------------------------------------LDA 5350
C                                                                      LDA 5360
C                                                                      LDA 5370
```

```
    1 FORMAT (2I5,I2,1P2E10.2,7E14.6/(32X,7E14.6))
    2 FORMAT (32X,1P7E14.6)
    3 FORMAT ('1  N =',I3,'   NL =',I3,'   RMSEPS =',1PE9.2,'   TEND ='
     1  ,E9.2,'   H =',E9.2//)
    4 FORMAT ('   NS    NW Q     H',8X,'T    ',8X,'Y(1,*) AND YL(*)'//)
      END
      SUBROUTINE COPYZ(S,Y,L)
C     ------------------------------------------------------------------COP   30
C                                                                       COP   40
C     THIS SUBROUTINE COPIES THE ARRAY Y, OF LENGTH L, INTO THE ARRAY S COP   50
C                                                                       COP   60
C     ------------------------------------------------------------------COP   70
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION S(1),Y(1)
      IF(L.LE.0)RETURN
      DO 100 J=1,L
  100 S(J) = Y(J)
      RETURN
      END
      SUBROUTINE DERVAL (Y,YL,T,N,NY,W,KERET)
C     THIS SUBROUTINE SUPPLIES THE INITIAL VALUES OF THE DERIVATIVES
C     OF THE NODAL PARAMETERS TAKEN FROM STEADY STATE SYSTEM ANALYSIS.
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION Y(7,1), YL(1), W(1)
      Y(2,7)=-9.428681D01
      Y(2,8)=-9.079442D01
      Y(2,9)=-9.428681D01
      Y(2,12)=-1.012731D02
      Y(2,13)=-2.894024D02
      Y(2,14)=-1.012731D02
      Y(2,17)=-1.057558D02
      Y(2,18)=-3.616863D01
      Y(2,19)=-1.057558D02
      Y(2,22)=-7.0622577D01
      Y(2,23)=-3.162976D02
      Y(2,24)=-7.062277D01
      Y(2,27)=-1.322337D02
      Y(2,28)=-3.825550D01
      Y(2,29)=-1.322337D02
      Y(2,42)=0.D0
      Y(2,43)=0.D0
      Y(2,44)=0.D0
      Y(2,47)=0.D0
      Y(2,48)=0.D0
      Y(2,49)=0.D0
      Y(2,52)=0.D0
      Y(2,53)=0.D0
      Y(2,54)=0.D0
```

```
      Y(2,57)=0.D0
      Y(2,58)=0.D0
      Y(2,59)=0.D0
      Y(2,62)=0.D0
      Y(2,63)=0.D0
      Y(2,64)=0.D0
      Y(2,88)=-4.157413D-01
      Y(2,89)=-3.571020D-01
      Y(2,90)=-3.156324D-01
      Y(2,91)=-3.571020D-01
      Y(2,92)=-4.157413D-01
      Y(2,93)=2.162937D-01
      Y(2,94)=2.371216D-01
      Y(2,95)=1.476146D-01
      Y(2,96)=2.371216D-01
      Y(2,97)=2.162937D-01
      Y(2,98)=-3.861831D-02
      Y(2,99)=-6.708371D-02
      Y(2,100)=-5.798223D-02
      Y(2,101)=-6.708371D-02
      Y(2,102)=-3.681831D-02
      Y(2,103)=1.207203D-01
      Y(2,104)=1.948975D-01
      Y(2,105)=2.023086D-01
      Y(2,106)=1.948975D-01
      Y(2,107)=1.207203D-01
      Y(2,108)=-2.532375D-01
      Y(2,109)=-2.842183D-01
      Y(2,110)=-3.355962D-01
      Y(2,111)=-2.842183D-01
      Y(2,112)=-2.532375D-01
      KERET=0
      RETURN
      END
      SUBROUTINE NUITSL (PW,DY,F1,N,NY,EPS,YMAX,NEWPW,KRET)
C     -----------------------------------------------------------------NUI   20
C     THE PURPOSE OF THIS SUBROUTINE IS TO SOLVE A                     NUI   30
C     LINEAR SYSTEM OF EQUATIONS FOR THE NEWTON ITERATES WHEN THE      NUI   40
C     CORRECTOR EQUATION IS BEING SOLVED.  UPON ENTRY TO THIS SUBROUTINENUI  50
C     THE SYSTEM OF EQUATIONS TO BE SOLVED IS   J W = -F  , WHERE      NUI   60
C        J  IS STORED IN PW UPON ENTRY                                 NUI   70
C        W  IS RETURNED IN F1                                          NUI   80
C        -F IS STORED IN DY UPON ENTRY                                 NUI   90
C                                                                      NUI  100
C     THIS SUBROUTINE IS GENERALLY SUPPLIED BY THE USER, ALTHOUGH THERE NUI 110
C     ARE SOME STANDARD FORMS AVAILABLE.  FOR EXAMPLE, THIS VERSION    NUI  120
C     ASSUMES THAT PW IS STORED IN FULL STORAGE MODE IN AN NXN MATRIX. NUI  130
C     IF NEWPW = 1, AN LU DECOMPOSITION IS DONE, NEWPW IS SET TO ZERO  NUI  140
```

```
C     AND FORWARD AND BACKWARD SUBSTITUTION FOR THE SOLUTION IS DONE.
C     IF NEWPW = 0, ONLY FORWARD AND BACKWARD SUBSTITUTION FOR THE
C     SOLUTION IS NECESSARY.
C
C     NOTE THAT THE PARAMETERS EPS AND YMAX ARE USEFUL IF AN ITERATIVE
C     METHOD IS USED TO SOLVE THE SYSTEM OF EQUATIONS.
C
C     THE CALLING SEQUENCE FOR THIS SUBROUTINE IS
C
C     CALL NUITSL(PW,DY,F1,N,NY,EPS,YMAX,NEWPW,KRET)
C
C     WHERE THE PARAMETERS ARE DEFINED AS FOLLOWS.
C
C        PW      - THE J MATRIX CALCULATED IN SUBROUTINE JACMAT
C        DY      - THE RIGHT HAND SIDE OF THE LINEAR SYSTEM TO BE SOLVED
C        F1      - THE SOLUTION IS RETURNED IN THE ARRAY F1
C        N       - SAME AS IN LDASUB, TOTAL NUMBER OF VARIABLES
C        NY      - SAME AS IN LDASUB, NUMBER OF DIFFERENTIAL EQUATIONS
C                  AND NONLINEAR VARIABLES
C        EPS     - L2 ERROR CONSTANT USED IN LDASUB
C        YMAX    - MAXIMUM VALUES OF Y(1,I) SEEN UP TO THE CURRENT TIME
C        NEWPW   - INDICATES WHETHER A NEW J MATRIX HAS BEEN COMPUTED
C                       =1  INDICATES A NEW J MATRIX HAS BEEN COMPUTED
C                           SINCE THE LAST ENTRY TO NUITSL.  NEWPW
C                           SHOULD BE SET TO ZERO IF SOME PREPROCESSING
C                           SUCH AS LU DECOMPOSITION MUST BE DONE ON A
C                           NEW  J  MATRIX.
C                       =0  INDICATES THE J MATRIX IS THE SAME AS WHEN
C                           NUITSL WAS LAST ENTERED
C        KRET    - RETURN INDICATOR
C                       =0  NORMAL RETURN
C                       =1  ERROR RETURN.  SOLUTION OF EQUATIONS COULD
C                           NOT BE OBTAINED.
C
C-----------------------------------------------------------------------
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION PW(1), DY(1), F1(1), YMAX(1)
      NL = N-NY
      IF (NEWPW.EQ.0) GO TO 100
      NEWPW = 0
      NN=N**2+1
      NNN=NN+N
      CALL LUDATF (PW,PW,N,N,0,D1,D2,PW(NN),PW(NNN),F1,IER)
      IF (IER.EQ.0) GO TO 100
      KRET = 1
      RETURN
  100 CALL LUELMF (PW,DY,PW(NN),N,N,F1)
      KRET = 0
```

```
      RETURN
      END
      SUBROUTINE DIFFUN(Y,YL,T,HINV,DY)
C     THIS SUBROUTINE EVALUATES THE SYSTEM'S GOVERNING SET OF
C     EQUATIONS AT A GIVEN TIME AND GIVEN VALUES OF THE NODAL
C     PARAMETER AND ITS DERIVATIVE.
      IMPLICIT REAL*8(A-H,O-Z,$)
      DIMENSION Y(7,1),YL(1),DY(1)
      COMMON CD(117,117),TM(117,117),C(117)
      DO 200 I=1,117
      DY(I)=-C(I)
      DO 100 J=1,117
  100 DY(I)=DY(I)+CD(I,J)*Y(2,J)*HINV+TM(I,J)*Y(1,J)
  200 CONTINUE
      RETURN
      END
      SUBROUTINE JACMAT(Y,YL,T,HINV,A2,N,NY,EPS,DY,F1,PW)
C     THIS SUBROUTINE EVALUATES THE JACOBIAN MATRIX AT THE GIVEN TIME
C     AND AT CURRENT VALUES OF THE DEPENDENT VARIABLES, ORDER,
C     AND STEPSIZE.
      IMPLICIT REAL*8(A-H,O-Z,$)
      DIMENSION Y(7,1),PW(117,1)
      COMMON CD(117,117),TM(117,117),C(117)
      AH=-A2*HINV
      DO 200 I=1,117
      DO 200 J=1,117
  200 PW(I,J)=TM(I,J)+AH*CD(I,J)
      RETURN
      END
```

LIST OF REFERENCES

1. Batchelor, G. K., "Heat Transfer by Free Convection Across a Closed Cavity Between Vertical Boundaries at Different Temperatures," Quarterly of Applied Mathematics, v. 12, p. 209-233, October 1954.

2. Cook, A. H., Interference of Electromagnetic Waves, p. 38-78, Ely House, 1971.

3. DeVahl Davis, G., "Laminar Natural Convection in an Enclosed Rectangular Cavity," Int. J. Heat Mass Transfer, v. 11, p. 1675-1691, 22 April 1968.

4. DeVelis, J. B. and Reynolds, G. O., Theory and Applications of Holography, p. 170, Addison-Wesley, 1967.

5. Eckert, E. R. G. and Carlson, W. O., "Natural Convection in an Air Layer Enclosed Between Two Vertical Plates with Different Temperatures," Int. J. Heat Mass Transfer, v. 2, p. 106-119, 22 January 1960.

6. Huebner, K. H., The Finite Element Method for Engineers, p. 67-118, p. 139-144, p.339-354, Wiley, 1975.

7. Kawahara, M., "Steady and Unsteady Finite Element Analysis of Incompressible Viscous Fluid," Int. J. Numerical Methods in Engineering, v. 10, p. 437-454, 16 January 1975.

8. Merzkirch, W., Flow Visualization, p. 102-184, Academic Press, 1974.

9. Newell, M. E. and Schmidt, F. W., "Heat Transfer by Laminar Natural Convection Within Rectangular Enclosures," J. Heat Transfer, v. 92, no. 1, p. 159-168, February 1970.

10. Oden, J. T., Finite Elements of Nonlinear Continua, p. 118-131, McGraw-Hill, 1972.

11. Schlichting, Hermann, Boundary-Layer Theory, p. 77, McGraw-Hill, 1968.

12. Smith, H. M., Principles of Holography, p. 186-193, Wiley-Interscience, 1969.

13. Steel, W. H., Interferometry, p. 75-95, Cambridge Univeristy Press, 1967.

14. Strang, G. and Fix, G. J., An Analysis of the Finite Element Method, p. 74-95, Prentice-Hall, Inc., 1973.

15. Szekely, J. and Todd, M. R., "Natural Convection in a Rectangular Cavity; Transient Behavior and Two Phase Systems in Laminar Flow," Int. J. Heat Mass Transfer, v. 14, p. 467-481, 18 June 1970.

16. Ural, O., Finite Element Method, Basic Concepts and Applications, p. 164-170, Intext Educational, 1973.

17. Zienkiewicz, O. C., The Finite Element Method in Structural and Continuum Mechanics, p. 148-166, McGraw-Hill, 1967.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 67 Department of Aeronautics Naval Postgraduate School Monterey, California 93940	1
4.	Professor D. J. Collins, Code 67 Co Department of Aeronautics Naval Postgraduate School Monterey, California 93940	2
5.	Dr. H. J. Mueller Naval Air Systems Command AIR 310 Department of the Navy Washington, D. C. 20360	1
6.	Dr. Charles M. Vest College of Engineering Department of Mechanical Engineering 550 East University Ann Arbor, Michigan 48109	1
7.	LT Gerald Paul Braun, USN 1441 Walbridge Avenue Toledo, Ohio 43609	1

Thesis 166684
B80189 Braun
c.1 Time dependent holographic interferometry and finite-element analysis of heat transfer within a rectangular enclosure.

Thesis 166684
B80189 Braun
c.1 Time dependent holographic interferometry and finite-element analysis of heat transfer within a rectangular enclosure.

thesB80189
Time dependent holographic interferometr
3 2768 002 07212 6
DUDLEY KNOX LIBRARY